

ATARI **COMPUTER** **ENTHUSIASTS**

3662 Vine Maple Dr. Eugene OR 97405

APRIL, 1985

Mike Dunn, Jim Bumpas & Larry Gold, Editors

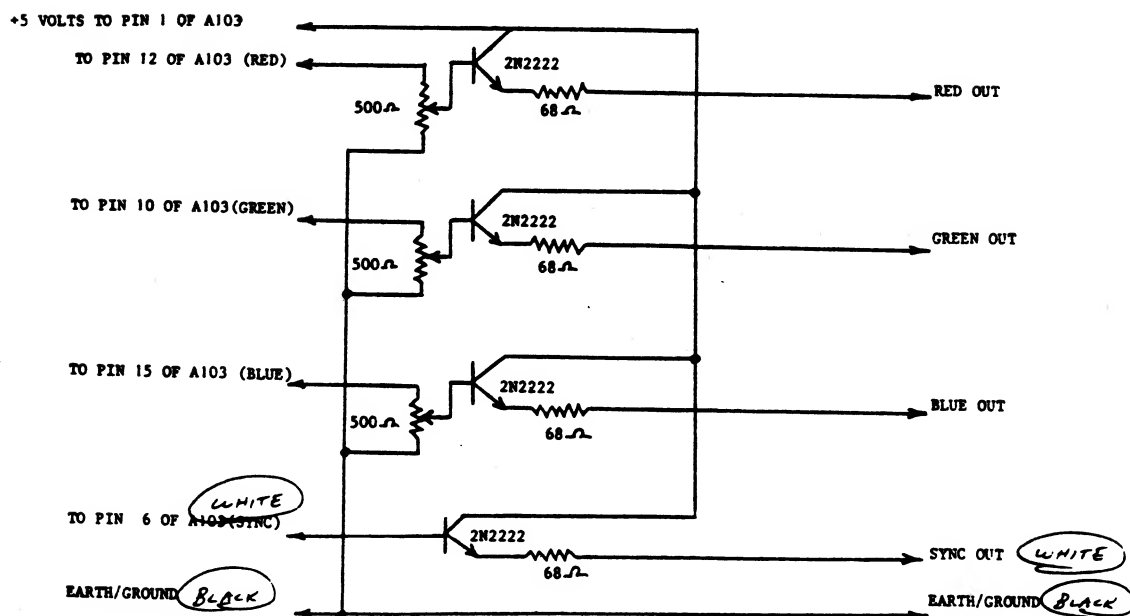


FIGURE 2 RGB OUTPUT FOR ATARI 800

News and Reviews

by Mike Dunn, Co-Editor

I am writing this article early, since I am going on vacation, so I don't know all of what's in this issue. There should be the great education program by Paul Freeman, **The Living Cell** and other fine articles and programs.

We have a number of new disks now ready. **Ace Telecommunications Disk #1** has KERMIT in run time versions for the 850 interface and modem of your choice. It also has KERMIT for the MicroBits Modem, and an intelligent modem program for the Atari 1030 modem. If you want KERMIT for the 1030 modem, let us know, and will be give you that version instead of the ones above. There was not enough room for the ACTION! source code; but if you really want it, specify and we will put in on the back of the disk. This disk would not have been possible without the aide of **MicroBits**, also known as **MPP** (225 Third Ave S.W., Albany, OR 97321) who put their fine programmers on the project to make a de-bugged run-time version of this program from the original one from Columbia University, written in ACTION! by John Pelevich, author of Chameleon. I understand his new version 4.2 is really great and available from ANTIC! for \$20. Microbits have been a fantastic supporter of ACE, not only helping us with programs, but supplying us with their Modems, interfaces, and other equipment to use, test, and keep.

There are also two new **Best of...** disks, both with a new DOS, MachDOS, given to ACE by B.A.S.I.C., the very nice disk-based newsletter from Brooklyn, NY. **Best of ACE #12** features a wonderful program from a German atari club called SKI that is much too long to put in the newsletter. Also features **SNAKES** and **The Living CELL** from this newsletter. **Best of ACE #13** has a public domain translator for the XL series that is much nicer than the "Official" Atari version, Solitaire in BASIC and in run-time C versions as well as some other run-time C programs by Ralph Walden, BlackHole, and programs from this issue.

The **ACE Business and DataBase** disk will be ready when we receive the list program by Stan Ockers, hopefully this issue.

Remember, any of the above are only \$10 each, or \$15 for 2 on a double sided disk. Write to Ron Ness for the above or any other disk you would want.

Another supporter of ACE from the beginning has sent us their just released Atari version of the best selling program, **The Print Shop** (Broderbund, \$45). This program will be reviewed by Larry Gold elsewhere, but it is really something you should consider buying if you have a graphic compatible printer — very easy to use, very impressive.

We hope to have access soon to a prototype version of the Atari ST — if so, a complete report next issue.

ATARIWRITER DRIVERS

Do you want to use one of the following printers with AtariWriter? If so, send \$10.00 to:

Gary Furr
Box 1073
Mountain View, CA 94042

Letter Quality:
Atari 1020, 1027
Compactronic 58
BMC PB 401
DTC Style Writer
Brother HR-15/25
Dot-matrix:
Epson MX, RX, & FX 80/100
Star Gemini 10/10X
Star Radix 10/15
Star SG,SR,SD - 10/15
Axon AT-100, SLP, GP550A
Mannesman Tally 160/180L, Spirit 80
Okidata 80,82/83,84,92/93
IDS 480 Microprism
C.Itoh Prowriter 8510
Olympia NP,RO,Comact2
Legend 880
CP-80 Type 1
Riteman
BMC BX-80
Olivetti PR2300 (ink)
Citizen MSP 10
Smith Corona D-100

Star Powertype
Diablo 620
Silver Reed 500-770
Juki 6100
Smith Corona TP-1

VP RAMBLINGS

It is my understanding that by the time you read this the 130XE computers should be in the stores. At what price your guess is as good as mine. I know alot of you will say so what, there is no software to take advantage of the extra memory. Wrong! There is some: most word processors should work, SynCalc, and most of the calc type programs, and the data base programs should also work well with this new machine. The market should see many types of programs to will take advantage of the added memory even if it is bank selected. Within a short time many of the programs around today can be enhanced to use more memory.

We have a new Bugbuster. He is DALE LUTZ, of Edmonton, Canada, and his phone no. is 403-688-2374. His specialty is general programming but with special emphasis on Action! All Atari users who have problems and are in his area you now have someone to call who can help you.

Anyone who has expertise in some area of computing and wants to become a Bugbuster please let me know and we will put you on our list.

Don't forget the contest for the best graphics which we will use for the newsletter and BBS.

— Larry Gold

THE PRINT SHOP

Print Shop (\$45, Broderbund) was first written for other computers, and then finally for the Atari. When I first saw it for the Apple I was very impressed and couldn't wait for it to come out for the Atari. Well, Broderbund should be very proud of themselves as they took a very, very good program and kept it that way. I was a little concerned that what was good on the Apple would be just so so on the Atari as has happened so many times in the past. This was not one of them.

The Print Shop is the ultimate user friendly program. (If only all the so called user friendly programs could be like this.) Menu driven, one can use it the minute you boot it up provided your printer is on. While the documentation is excellent it really isn't needed.

I booted the program up and as I went through the menu I made two birthday cards and some letterheads for some stationary without any trouble at all and without the aid of any instructions. Not bad for someone who can't use most programs without reading the instructions at least twice.

With the program comes various types of paper. More can be had as Broderbund has brought out a packet with paper of different types, and I understand they are bringing out more fonts and graphics for the user.

With this program one can make cards, letterheads, banners, etc. You can let your imagination run free as to what it can do. It utilizes graphics made on the Koalapad, Atari Touch Tablet and the various graphic programs which are available. This is truly a program everyone can use who has a printer. Anyone from 8 to 80 will find it a program which is fun to use and which makes a finished product which is a joy to behold.

The first menu offers the selection of the type of job to do. The next menu permits selection of border types (thick, thin, double, light, dark). Next you can add graphics (your own or "canned" ones in the program). Then you select layout options, including size, type of font, and how many you want. You then put in your custom message, letter, or what-have-you. Finally to the print routine and here Print Shop supports 14 types of printers, and more I suppose when they come out.

I can only say that here is one of the best programs written for any computer and one of the best programs one can own. I only wish it was written for the Atari first so we could have had it that much sooner. Broderbund is to be commended and I hope they keep up the good work and bring us more programs to do what they are intended to do and with such ease. With programs such as The Print Shop, computing is really a joy and also fun which makes it that much better for everyone.

— Larry Gold

PITCH RECALL

Here is a game to test your ability to distinguish pitch of tones. Either one or two people may play. Use SELECT to toggle between one or two players at the beginning of the game. The difficulty for each player may be set by pressing the option key. Player 1 uses a joystick in port 1 and player 2 one in port 2. You will be asked to memorize a tone and then recall it by using the joystick to match the tone you heard. A response coming within certain limits of the original tone will be rewarded by adding a note to your score. Ten notes wins the game. The game can be restarted by pressing START while the tones are alternating at the end of a turn.

— Stan Ockers

BUMPAS REVIEWS

KAMPFGRUPPE (\$60, SSI, 883 Stierlin Rd. Bldg A-200, Mountain View, CA 94043) is the latest strategy game by Gary Grigsby. This 2-disk game simulates tactical combat between Russia and Germany from 1941 to 1945. Maneuver elements are platoons and batteries. One or two may play, or you may select the Atari to play both sides.

The Game disk provides 4 historical scenarios, set up and ready to play: Meeting Engagement East of Bryansk (7-7-42); Stalingrad (12-17-42); Kiev (11-3-43); and, Berlin (3-22-45). The last three scenarios are all assaults against prepared positions. Don't be disappointed if the game is not much of a challenge for solitaire play. My first two games were the Stalingrad scenario, and I scored a substantive victory on each side.

Mr. Grigsby has contributed a great deal of imagination and creativity to produce a strategy game with even more flexibility than the traditional paper board game. In the 2-player mode, this game really shines.

The opening menu permits a handicap level for the current game. In addition to the 4 prepared scenarios there is the choice to build your own. This is the most powerful feature of this game. Its implementation here is the most elegant and powerful I've seen for any computer. If you choose, the Atari will draw the map, select the forces for the two sides, and deploy them for you. You can set up an entirely original scenario in a matter of seconds.

If you want to take a little longer, you may deploy your own forces. A little more time and you may select the forces used. First you select either a Tank or Infantry type force (Russian); or, Panzer, Motorized or Infantry (German). Then you choose which battalions, companies and detachments you will have from a screen menu. Then when you deploy, you may substitute armored vehicles from a data card (which comes with the game) for those vehicles assigned to the formations you chose. For instance, if the tank company you chose contains BT-7s, you can substitute T34s. You have 8 time periods from which to choose, and you cannot select a vehicle which was not available in the time period selected.

If you really want to get your creative juices flowing, you can design the map upon which you will play. The Atari will draw the map, and will include a river if you like. It will also ask you to control the quantity of woods and broken terrain on the map. When you deploy, you may press "U" on the keyboard. This act opens the door to selecting from among 59 terrain types to place in the space under the cursor. Possible terrain includes: clear, road, town, woods, broken, slope, road/slope, river, bridge, and ford. These terrain types may be placed upon 7 different topographical levels (the line-of-sight rules are very sophisticated). The map is a 60x60 grid scrolling over approximately 18 screens, so creating a map is a big job. Each space is meant to represent 200 yards square. But you may make it just as you want it! The player control over this game is fantastic, and makes it my favorite game.

Players also choose the scenario type from among the following choices: Pursuit, Assault, or Meeting Engagement. And you may choose the size of the battle: Large, or Small. Large battles have twice as many units as Small ones.

In play, the screen shows to each player only those enemy units visible to his units. There is a View command which will paint in orange all the spaces a unit can see. Another command will list each target a unit can see.

Units may be set in "close combat mode". These units will move in to the space of a firing enemy unit within 3 spaces before continuing to its commanded objective.

All commands are by keyboard, and certain commands may be given to all units in a formation. Trucks and halftracks provide transportation to guns, mortars and troops. Additionally, infantry may become "Tank Marines" and ride on tanks. The Atari keeps track of command control, and you can trace the chain of command for any unit by pressing "H" until the command HQ is reached. Units do not execute commands immediately — there is always some delay. Loss of command control extends the delay.

Combat execution takes into account the aspect of the armored target (front armor is harder to penetrate). Other elements in the combat algorithms include: weapon accuracy, number of times fired at target, defensive terrain, target movement, target silhouette, weapon penetration, number of weapons shooting at same target, and the suppression level of the target. Ammunition is expended each turn, and I always seem to run out of it before the scenario is over. But armored vehicles may still overrun targets without ammunition. It's not as effective as with ammunition, but hey! If it's all you've got, you use it.

The Atari scores the game. You might achieve a draw, or a marginal, substantive, or decisive victory.

The 24 pages of documentation include maps and descriptions of the 4 scenarios. There are 5 pages of boxes containing capsule weapon descriptions and silhouettes, and a page of strategy notes. There is also a BASIC program listing with which you may test out direct fire probabilities against armored targets. Finally, there is a 5-page combat short story.

This game will probably have more play value in it than any previous strategy game. You probably will not live long enough to play all the possible combinations of scenarios possible with this game. **Kampfgruppe** is getting very close to a strategy game generator for those with the creativity to develop new games but who lack the programming skill to translate their ideas to bits and bytes. I hated to put aside Gary Grigsby's **War in Russia** game, but this one has to be played.

MOSAIC + SYN-APPS

A user, Charlie Cottle, reports to me he has a 165k data buffer available when he uses SynCalc. He has 184k RAM available when he uses SynFile+. Is he running these programs on an IBM? Nope, he has an Atari 800 with three 64k Mosaic RAM boards. If you have an 800, and are thinking maybe you want to have the extra memory of a 130XE machine, you can get 3 Mosaic boards and have 64k more than an XE.

TRIVIA QUEST (\$40, Royal Software) is a 4-disk social game for those who want more than plain-vanilla trivia for 4 players, each with a joystick. XE and XL owners are limited to 2 players. If you also want (and expect!) excellent animated graphics and a little arcade-style action, then this package from our friends here in Eugene is for you.

The hand of John Atack is obvious in the high-resolution graphics and animation. You know him from Quasimoto. The graphic design shows a trail of boxes winding through a terrain of woods, mountains, rivers and castles scrolling over several screens.

Each player gets individual questions to answer. The quicker the answer the more food is gained. Food is important because with more food one can travel further toward the goal of "Trivia Quest". This goal is to march your questing party of three characters (Prince, Knight, and Page) around the board and back to the castle again. Each box upon which you stop may determine the type of question one is asked. Royal Software provided us only with a dealer demo disk, so I cannot comment on the quality of the questions. And while there are categories, the players do not control them.

Occasionally a box will contain a dragon. If you defeat the dragon with arrows, you win all his gold. If the dragon defeats you, it will take some of your gold. Gold is also won by answering questions. The higher the difficulty, the more gold is in the prize. You need gold to outfit your expedition — the Knight and Prince require more gold to outfit than the Page.

You can challenge another player to a question-off if you press the joystick trigger as you pass through a space containing a character, you may both go to a question screen where only the first one answering wins the gold and food. Answers are chosen by moving the joystick among 4 choices for each question. The response is very fast.

Royal Software is now preparing a utility disk which will permit you to create and edit your own-design question disks. This can be a valuable home educational tool — especially for those who may be motivated by video games. They are promising this disk in 10 days, so it should be ready by the time you read this.

— Jim Bumpas

BIT TWIDDLING

(reprint: StarFleet, Feb. 1985)

In the past, I have been frequently exasperated by the lack of BASIC commands to do bit manipulations. The normal BASIC logical operations return a one or zero (true or false). Therefore, you usually test a variable with a logical operation, then use arithmetic operations to set or mask specific bits you wish to change.

Aside from being slow, the code tends to be very cryptic. Three months later, you may end up rewriting the whole program rather than try to figure out what you did.

The following routines are a solution to the problem. They allow true, direct bit manipulation of 16 bit integers by means of a single USR call. The AND and OR allow bit masking and the EXCLUSIVE OR may be used for certain types of encrypting schemes.

The necessary initialization sections are included in the main program and called to set up a small machine language routine which is stored in a string variable. The machine language routine is called by the main program as shown in the example. Two variables (arguments) ARG1 and ARG2 are passed within the USR command. The USR command is made equal to the result variable. The string variables have been given names which identify the purpose of the operation (e.g., LAND = logical AND) to make debugging easier. I hope these are as useful to you as they have been to me.

— Mike Cunningham

BIG

BIG is a PILOT program written as a learning game for handicapped children. It is designed to teach size value judgments and improve basic computer literacy. The child uses a number select menu to pick out a program or (if successful) a game. It only requires the numbers 1-6 be inputted and return pressed for each response.

WHY BOTHER WITH THE HANDICAPPED?

Attitudes and preconceived notions have lessened the impact of one of the most rewarding uses of the micro computer — helping mentally handicapped children and adults. As an instructor of moderate to profoundly retarded children I have found it necessary to create useful programs.

Chosen because of its easy to use personality, the Atari computer and the PILOT language have made it easier to write software. Many publishers of educational materials claim their materials work with our students, but this is NOT true. Watered down public school special education lessons are not effective for our disabled learners. With regular materials such as books, puzzles, and records we are at least able to use and reuse the useful parts. Most commercial software is either protected or too complex to allow the teacher to modify it for the handicapped student.

Our handicapped learners are not as different as some people believe. They do need more repetition and easier levels to absorb a new concept or activity. Daily use of the computer for 2 years has shown us transfer (relating something learned in one area to another area) does occur. There are many myths about the retarded, but two major points are real. Retardation does not go away.

It is a life-long condition. The child can learn, but only with some special help. The computer can continue as a learning aid not only for children but for adults of any age.

Retarded individuals can and do live useful and productive lives with just a little assistance. In our county, a sheltered workshop works hand-in-hand with local industries providing gainful employment through subcontract work. A group of individuals who could have been inmates of institutions at the taxpayers' expense are now productive taxpayers themselves.

A DIFFERENT HANDICAP

The child with a mental handicap needs individualized programs to be able to use the computer appropriately. The range of intellect in a single classroom of students (aged 13-18, for example) is as varied as might be found in a one room schoolhouse serving all 12 grades. We also have a significant number of students with no speech or multiple difficulties such as cerebral palsy.

We began using the computer with an easy to use commercial program for pre-schoolers using a joystick and only requiring the matching of shapes and letters. Soon most of our children were using the keyboard. The Atari 400 provides a nice input device when modified by placing paper over the keys not wanted. Learning to use the computer was easy. The randomness of the required responses keep the students from memorizing the answers. Because of the need for repetition and he need for individualized attention (especially feedback for responses), many special educators who have never used a computer are using techniques easily done by the computer.

THE PROGRAM

The PILOT program BIG will be interesting to preschoolers as a game. It uses several features I feel are necessary for educational software:

1. Only correct responses are rewarded. No negative noise or action occurs upon an incorrect response. Every correct response has a tiny reward.
2. The required responses are understood and possible. Success needs to occur before learning can happen.
3. Learning steps are small enough to be possible and large enough to hold interest. For our special learners, the steps are slower and more repetitive but they are still there.
4. Fun may be an essential ingredient. The enthusiasm of the individual towards an activity often determines success. While all lessons cannot be fun, they can at least be interesting and enjoyable.
5. This program, although simple in concept, not only provides practice in size and value judgments, but also improves literacy by using a number select menu to allow the student (or teacher) to pick the program and also (if successful) to choose the reward.

The first program module, Biggest, uses the turtle to draw four unequal red vertical lines. The text window displays the numbers 1 to 4 under the lines from left to right. The student picks out the correct line, touches the line on the screen and bring the finger down to the number underneath. The number is then pressed on the keyboard. Finally, after checking to see if the proper number is in the window, the return key is pressed.

The second lesson module, Bigger, uses the turtle to draw two horizontal yellow lines of unequal length. The student is requested to press the correct number key. The student must then press the return key.

Measure, the fifth module, has the turtle draw one horizontal line at random from 1 to 6 inches long. The student places a homemade plastic coated paper ruler with large one inch markings under the line of the screen. Since the object of the program is to provide successful drill, the line always starts at the same place, but jumps randomly to one of 5 lengths to insure the child has practice positioning the ruler. More advanced students can also use this program to guess the size.

A short pleasant sound occurs upon a correct response. Upon an incorrect response, the prompt "try again" appears. I do not want to reinforce the incorrect response, but I need some indicator the answer was wrong. A counter in the program records all incorrect responses. When the student is finished, the score is displayed. Even though there is a short sound reward for each correct response and a score printout at the end of ten responses, a special reward appears for 10 correct responses in a row. The successful student is presented with a screen menu of reward choices. The selection includes a drawing of a picture and drawing a variety of shapes and colors according to the student's directions to the turtle.

I will send you a copy of this program on your tape or disk for \$3 and a SASE.

— Carl Schwartz
5607 Scribner Road
Pierpont, OH 44082

LABELS UPDATE

I find it's easier to remove the tractor feed from the Gemini 10X and print labels directly on individual envelopes. To help in this I made the following changes to the program:

1) add line 401: 401 POSITION 3,18: ? " HOW MANY BLANKS BEFORE LABEL ":INPUT MARG

2) change line 406: 406 ? #2:BLK\$(1,MARG);

3) change the ? #2: " " ; in line 412 to ? #2:BLK\$(1,MARG);

Answer the number of blanks question with:

A) 6 for labels

B) 24 for regular envelopes

C) about 40 for business size envelopes

I also find the program useful for printing disk labels. If you are setting up the program in anticipation of printing lists, I find I should have filled the index with the last letter of the alphabet rather than with X's (the small letter 'z'). The changes will be in lines 5010 and 5020:

1) in 5010 XSS\$ = "z"

2) in 5020 FREX\$ = "zzzzz"

— Stan Ockers

BLACKBIRD

BLACKBIRD is a space game which includes fast action, color, sound, and nine different screens! In the game, you must maneuver your spacecraft through a winding series of caverns, avoiding the deadly fire of surface-to-air missiles. You may fight back by dropping photon bombs (press the trigger). You accumulate points by hitting the missile launchers and finishing each screen. If you make it to the last screen, you are declared a winner, but chances are you won't make it that far!

The animation in BLACKBIRD is fast and smooth, even though the program is written completely in BASIC. To achieve the graphics in BLACKBIRD, I used a number of advanced techniques, such as Player/Missiles and a custom character set. In order to have smooth vertical movement of the player/missiles, strings are used. In many cases, it is better to use strings (instead of machine-language subroutines) to move P/M graphics. The string handling functions which can be used in BASIC allow the user to move large areas of memory in a short time, and this speed can be used to great effect with player/missiles.

The main screen display is in graphics mode 1 (this saves a lot of memory). Most of the characters of the alphabet were redefined to form sections of the cave. By using character graphics instead of mode 7 or 7.5, it is possible to draw a new screen in a very short time. BLACKBIRD requires at least 40K and BASIC. Grab a joystick and have fun!

— Paul Freeman

THE CELL

BY PAUL FREEMAN

```

2 REM THE CELL
3 REM Paul Freeman
4 REM 2-14-85

40 CLR :DEG :POKE 82,0:GRAPHICS 7:SETC
OLOR 4,0,0:SETCOLOR 2,0,4:SETCOLOR 1,1
3,10:SETCOLOR 0,3,8:POKE 752,1
50 CH=80:CV=52:DIM B0(14),DA$(45),AN$(
5)
70 COLOR 3:PLOT 6,0:DRAWTO 152,0:COLOR
1:PLOT 6,2:DRAWTO 152,2:DRAWTO 152,78
:DRAWTO 6,78:DRAWTO 6,2
75 FOR A=8 TO 150 STEP 2:PLOT A,3:PLOT
A,77:PLOT A,4:PLOT A,76:NEXT A
77 FOR A=4 TO 76 STEP 2:PLOT 7,A:PLOT
151,A:PLOT 8,A:PLOT 150,A:NEXT A
90 RESTORE :COLOR 2:FOR A=10 TO 41:REA
D B:PLOT (80-B)-B*0.5,A:DRAWTO (80+B)+
B*0.5,A:PLOT (80-B)-B*0.5,78-A
96 DRAWTO (80+B)+B*0.5,78-A:SOUND 0,90
-A*2,10,3:NEXT A:SOUND 0,0,0
100 DATA 5,10,12,15,16,18,19,20,21,22,
24,25,25,26,26,27,28,28,29,29,30,30,31
,31,32,32,32,32,33,33,33,33
110 RESTORE :COLOR 1:FOR A=10 TO 41:RE
AD B:PLOT (78-B)-B*0.5,A:PLOT (79-B)-B
*0.5,A:PLOT (82+B)+B*0.5,A
120 PLOT (78-B)-B*0.5,78-A:PLOT (82+B)
+B*0.5,78-A:PLOT (81+B)+B*0.5,A:PLOT (
79-B)-B*0.5,78-A:PLOT (81+B)+B*0.5,78-
A:NEXT A
130 COLOR 3:FOR A=38 TO 43:READ B:PLOT
80-B,A:DRAWTO 80+B,A:PLOT 80-B,86-A:D
RAWTO 80+B,86-A:NEXT A
140 DATA 1,3,4,4,5,5
150 RESTORE 140:COLOR 0:FOR A=38 TO 43
:READ B:PLOT 79-B,A:PLOT 81+B,A:PLOT 7
9-B,86-A:PLOT 81+B,86-A:NEXT A
160 COLOR 1:PLOT 60,30:DRAWTO 66,22:DR
AWTO 63,21:DRAWTO 58,29:DRAWTO 60,30:P
LOT 60,27:DRAWTO 63,28:PLOT 62,24
165 DRAWTO 65,25
170 PLOT 95,25:DRAWTO 99,33:DRAWTO 102
,32:DRAWTO 99,25:DRAWTO 95,25:PLOT 99,
25:DRAWTO 95,26
180 COLOR 3:PLOT 56,50:DRAWTO 58,56:DR
AWTO 64,58:DRAWTO 70,57:PLOT 58,52:PL
O 56,54:PLOT 64,59:PLOT 68,56
200 PLOT 85,53:DRAWTO 84,55:DRAWTO 86,
58:DRAWTO 93,62:PLOT 88,52:DRAWTO 87,5
4:PLOT 89,56:DRAWTO 94,58
210 PLOT 90,52:PLOT 91,52:PLOT 90,53:P
LOT 91,53:PLOT 94,54:PLOT 95,54:PLOT 9
4,55:PLOT 95,55
230 ? " --- BY PAUL FREEMAN"

3 N ---"
240 ? " Move Joystick To Make Selec
tion. "?:
260 COLOR 0:FOR A=0 TO 6:PLOT 55,A:DRA
WTO 105,A:NEXT A
270 COLOR 1:PLOT 59,0:DRAWTO 63,0:PLOT
61,1:DRAWTO 61,6:PLOT 65,0:DRAWTO 65,
6:PLOT 66,3:DRAWTO 68,3:PLOT 69,0
280 DRAWTO 69,6:PLOT 75,0:DRAWTO 71,0:
DRAWTO 71,6:DRAWTO 75,6:PLOT 72,3:DRA
WTO 74,3
290 PLOT 83,0:DRAWTO 79,0:DRAWTO 79,6:
DRAWTO 83,6:PLOT 89,0:DRAWTO 85,0:DRA
WTO 85,6:DRAWTO 89,6:PLOT 86,3
300 DRAWTO 88,3:PLOT 91,0:DRAWTO 91,6:
DRAWTO 95,6:PLOT 97,0:DRAWTO 97,6:DRA
WTO 101,6
310 FOR A=15 TO 0 STEP -1:SETCOLOR 4,0
,A:FOR M=1 TO 10:NEXT M:NEXT A
320 COLOR 3:RESTORE 350:FOR A=0 TO 16:
READ B,C:PLOT B,C:FOR D=0 TO 6:READ B,
C:IF B=-1 OR C=-1 THEN NEXT A:GOTO 401
330 DRAWTO B,C:NEXT D:NEXT A:GOTO 401
350 DATA 40,74,40,70,42,70,42,72,41,72
,-1,-1,44,74,44,70,46,70,46,72,44,72,4
6,74,-1,-1,50,74,48,74,48,72,49,72
360 DATA 48,72,48,70,50,70,-1,-1,52,74
,54,74,54,72,52,72,52,70,54,70,-1,-1,5
6,74,50,74,50,72,56,72,56,70,58,70
370 DATA -1,-1,62,74,64,74,64,72,62,72
,62,70,64,70,-1,-1,67,74,67,70,68,70,6
6,70,-1,-1,70,74,70,70,72,70,72,74
380 DATA 72,72,71,72,-1,-1,74,74,74,70
,76,70,76,72,74,72,76,74,-1,-1,79,74,7
9,70,80,70,70,70,-1,-1,84,74,84,72
390 DATA 85,72,84,72,84,70,86,70,-1,-1
,88,74,88,70,90,70,90,74,88,74,-1,-1,9
2,74,92,70,94,70,94,72,92,72,94,74
395 DATA -1,-1,100,74,100,70,98,70,98,
74,101,74,-1,-1,103,70,103,74,105,74,1
05,70,-1,-1,107,74,107,70,-1,-1
396 DATA 109,70,112,70,109,74,112,74,-
1,-1
400 REM SELECTION LOOP
401 OST=ST:ST=STICK(0)
402 COLOR 2:PLOT CH,CV:PLOT CH+1,CV:PL
OT CH,CV-1:PLOT CH+1,CV-1
403 IF PEEK(53279)=6 THEN GOTO 7000
405 IF ST=14 THEN CH=80:CV=25
410 IF ST=6 THEN CH=105:CV=26
415 IF ST=7 THEN CH=128:CV=40
420 IF ST=5 THEN CH=100:CV=52
425 IF ST=13 THEN CH=80:CV=52
430 IF ST=9 THEN CH=72:CV=54
435 IF ST=11 THEN CH=65:CV=50
440 IF ST=10 THEN CH=66:CV=30
450 IF ST<15 THEN SOUND 0,(CH+CV)/2,1
0,5
455 IF ST=15 THEN SOUND 0,0,0,0
500 IF CO<5 THEN COLOR 0:PLOT CH,CV:PL
OT CH+1,CV:PLOT CH,CV-1:PLOT CH+1,CV-1
505 IF CO=5 THEN COLOR 2:PLOT CH,CV:P
LOT CH+1,CV:PLOT CH,CV-1:PLOT CH+1,CV-
1
510 CO=CO+1:IF CO>10 THEN CO=0
520 IF OST=ST THEN GOTO 590
525 IF ST=15 THEN GOTO 590
530 ? " Move Joystick To Make Select
ion. "
540 IF ST=14 THEN ? " CYTO
PLASM"
545 IF ST=6 THEN ? " CENTR
IOLE"
550 IF ST=7 THEN ? " CELL M
EMBRANE"
555 IF ST=5 THEN ? " GOLGI
BODY"
560 IF ST=13 THEN ? " N
UCLEUS"
565 IF ST=9 OR ST=11 THEN ? "
ENDOPLASMIC RETICULUM"
575 IF ST=10 THEN ? " MITOC
HONDRIA"
580 ? " Press Button To View Select
ion. "
590 IF STRIG(0)=1 THEN GOTO 400
600 COLOR 0:FOR A=10 TO 74:PLOT 25,A:D
RAWTO 135,A:SOUND 0,A,10,3:NEXT A
620 IF CH=80 AND CV=52 THEN GOSUB 4000
630 IF CH=105 AND CV=26 THEN GOSUB 430
0
640 IF CH=66 AND CV=30 THEN GOSUB 4600
650 IF CH=100 AND CV=52 THEN GOSUB 480
0
660 IF (CH=72 AND CV=54) OR (CH=65 AND
CV=50) THEN GOSUB 5000
670 IF CH=80 AND CV=25 THEN GOSUB 5200
680 IF CH=128 AND CV=40 THEN GOSUB 540
0
700 GOTO 90
4000 REM NUCLEUS
4002 RESTORE :COLOR 3:FOR A=10 TO 41:R
EAD B:PLOT 80-B,A:DRAWTO 80+B,A:PLOT 8
0-B,78-A:DRAWTO 80+B,78-A
4003 SOUND 0,90-A*2,10,3:NEXT A:SOUND
0,0,0,0
4010 RESTORE :COLOR 1:FOR A=10 TO 41:R
EAD B:PLOT 78-B,A:PLOT 79-B,A:PLOT 81+

```

THE CELL CONT

```

B,A:PLOT 82+B,A:PLOT 78-B,78-A
4015 PLOT 79-B,78-A:PLOT 81+B,78-A:PLO
T 82+B,78-A:NEXT A
4020 RESTORE 140:COLOR 2:FOR A=42 TO 4
7:READ B:PLOT 80-B,A:DRAWTO 80+B,A:PLO
T 80-B,94-A:DRAWTO 80+B,94-A:NEXT A
4030 RESTORE 140:COLOR 0:FOR A=42 TO 4
7:READ B:PLOT 79-B,A:PLOT 81+B,A:PLOT
79-B,94-A:PLOT 81+B,94-A:NEXT A
4040 COLOR 1:PLOT 79,47:DRAWTO 81,47:P
LOT 79,48:DRAWTO 81,48:PLOT 80,46:PLOT
80,49
4050 FOR A=1 TO 50:B=INT(60*RND(1))+50
:C=INT(60*RND(1))+10:LOCATE B,C,D
4060 IF D=3 THEN GOTO 4070
4065 NEXT A:GOTO 4090
4070 COLOR 1:PLOT B,C:PLOT B+1,C:PLOT
B,C+1:PLOT B+1,C+1:NEXT A
4090 ? " The NUCLEUS is a membrane bo
und":? " structure which controls the
"
4095 ? " activities of the cell. It c
ontains ":FOR M=1 TO 100+DEM*600:NEXT
M
4100 IF STRIG(0)=1 AND DEM=0 THEN GOTO
4100
4110 ? " the chromosomes, which carry
the":? " genetic information. The nu
cleus"
4115 ? " usually contains two smaller
bodies":FOR M=1 TO 100+DEM*600:NEXT M
4120 IF STRIG(0)=1 AND DEM=0 THEN GOTO
4120
4130 ? " called nucleoli, that have ri
bonucleic":? " acid (RNA).":?
4135 FOR M=1 TO 100+DEM*600:NEXT M
4140 IF STRIG(0)=1 AND DEM=0 THEN GOTO
4140
4150 COLOR 0:FOR A=8 TO 72:PLOT 40,A:D
RAWTO 120,A:SOUND 0,A,10,3:NEXT A:SOUN
D 0,0,0,0
4160 IF DEM=1 THEN GOTO 15010
4170 RETURN
4300 REM CENTRIOLE
4302 SOUND 0,0,0,0:RESTORE 4310:COLOR
1:PLOT 20,35:FOR C=35 TO 65:READ B:DRA
WTO B+20,C:NEXT C
4310 DATA 0,0,0,0,0,0,0,0,1,1,1,2,2,
3,3,4,4,5,5,6,7,8,9,10,11,12,13,15,17,
20,21
4320 RESTORE 4310:PLOT 100,15:FOR C=15
TO 45:READ B:PLOT B+100,C:NEXT C
4330 RESTORE 4310:PLOT 121,45:FOR C=45
TO 15 STEP -1:READ B:DRAWTO 121-B,C:D
RAWTO 122-B,C:NEXT C
4340 RESTORE 4310:FOR C=35 TO 65 STEP
2:COLOR 1:READ B:PLOT B+21,C:DRAWTO B+
99,C-20:COLOR 3:READ B:PLOT B+21,C+1
4345 DRAWTO B+99,C-19:SOUND 0,117-C,10
,3:NEXT C:SOUND 0,0,0,0
4400 ? " Two CENTRIOLES are found in
all":? " animal cells. Each centriole
is a"
4405 ? " cylinder of nine sets of thr
ee":FOR M=1 TO 100+DEM*600:NEXT M
4410 IF STRIG(0)=1 AND DEM=0 THEN GOTO
4410
4420 ? " microtubules parallel to the
":? " cylindrical axis. They control
the"
4425 ? " activities of the mitotic sp
indle":FOR M=1 TO 100+DEM*600:NEXT M
4430 IF STRIG(0)=1 AND DEM=0 THEN GOTO
4430
4440 ? " and serve as a base for the
cilia":? " and flagella":? :FOR M=1 T
O 100+DEM*600:NEXT M
4450 IF STRIG(0)=1 AND DEM=0 THEN GOTO
4450
4460 COLOR 0:FOR A=8 TO 72:PLOT 15,A:D
RAWTO 145,A:SOUND 0,A,10,3:NEXT A:SOUN
D 0,0,0,0
4470 IF DEM=1 THEN GOTO 15010
4480 RETURN
4600 REM MITOCHONDRIA
4602 COLOR 2:RESTORE 4610:PLOT 40,60:F
OR A=0 TO 9:READ B,C:DRAWTO B,C:NEXT A
4610 DATA 46,70,60,68,86,58,112,32,120
,20,112,10,102,12,74,28,56,36,40,60
4615 SOUND 0,0,0,0
4620 COLOR 2:RESTORE 4610:PLOT 40,61:F
OR A=0 TO 9:READ B,C:DRAWTO B,C+1:NEXT
A
4630 RESTORE 4635:COLOR 3:PLOT 41,61:F
OR A=0 TO 9:READ B,C:DRAWTO B,C:NEXT A
4635 DATA 46,68,60,66,86,56,112,30,119
,20,111,12,101,14,74,30,57,38,42,60
4640 COLOR 3:RESTORE 4650:FOR A=0 TO 9
:READ B,C,D,E:PLOT B,C:DRAWTO D,E:NEXT
A
4650 DATA 102,15,118,22,92,21,108,28,8
6,25,106,38,82,27,90,40,76,35,88,46,66
,35,90,52,64,43,86,58
4655 DATA 58,43,66,54,50,49,60,60,46,5
5,60,68
4660 COLOR 1:RESTORE 4670:FOR A=0 TO 7
:READ B,C:PLOT B,C:PLOT B,C+1:PLOT B+1
,C:PLOT B+1,C+1:NEXT A
4670 DATA 102,18,104,32,88,30,92,46,70
,40,74,54,58,48,52,66
4680 FOR A=60 TO 20 STEP -1:SOUND 0,A,
10,3:NEXT A:SOUND 0,0,0,0
4690 ? " MITOCHONDRIA are present in t
he":? " cytoplasm of all aerobic cel
ls,"
4700 ? " They consist of an outer mem
brane and":FOR M=1 TO 100+DEM*600:NEXT
M
4705 IF STRIG(0)=1 AND DEM=0 THEN GOTO
4705
4710 ? " an inner membrane with many
bends":? " for a higher surface area.
The"
4720 ? " high energy molecule adenosine":FOR M=1 TO 100+DEM*600:NEXT M
4725 IF STRIG(0)=1 AND DEM=0 THEN GOTO
4725
4730 ? " triphosphate (ATP) is formed
in":? " structures attached to the i
nner"
4740 ? " membrane of the mitochondria.
":FOR M=1 TO 100+DEM*600:NEXT M
4745 IF STRIG(0)=1 AND DEM=0 THEN GOTO
4745
4750 COLOR 0:FOR A=8 TO 72:PLOT 40,A:D
RAWTO 120,A:SOUND 0,A,10,3:NEXT A:SOUN
D 0,0,0,0
4760 IF DEM=1 THEN GOTO 15010
4770 RETURN
4800 REM GOLGI BODY
4802 COLOR 3:RESTORE 4810:FOR A=0 TO 6
:READ B,C,D,E,F,G:PLOT B,C:DRAWTO D,E:
DRAWTO F,G:NEXT A
4810 DATA 32,38,60,22,80,18,36,50,54,3
4,70,24,48,46,72,30,86,30,54,48,78,36,
92,44,92,36,100,46,104,60,74,22
4815 DATA 92,26,106,46,84,18,104,28,12
0,50
4818 SOUND 0,0,0,0
4820 COLOR 3:RESTORE 4810:FOR A=0 TO 6
:READ B,C,D,E,F,G:PLOT B,C+1:DRAWTO D,
E+1:DRAWTO F,G+1:NEXT A
4830 COLOR 2:RESTORE 4840:FOR A=0 TO 3
:READ B,C:PLOT B,C:DRAWTO B+2,C:PLOT B
-1,C+1:DRAWTO B+3,C+1:PLOT B,C+2
4835 DRAWTO B+2,C+2:PLOT B+1,C-1:PLOT
B+1,C+3:NEXT A
4840 DATA 56,56,80,48,92,62,70,50
4900 ? " GOLGI BODIES are cytoplasmic
":? " organelles which consist of a s
tack"
4910 ? " of flattened tubules surroun

```

THE CELL CON'T

```

ded by":FOR M=1 TO 100+DEM*600:NEXT M
4915 IF STRIG(0)=1 AND DEM=0 THEN GOTO
4915
4920 ? " spherical objects. The golgi
bodies":? " have the primary role of
complexing"
4930 ? " proteins formed in the endop
lasmic":FOR M=1 TO 100+DEM*600:NEXT M
4935 IF STRIG(0)=1 AND DEM=0 THEN GOTO
4935
4940 ? " reticulum with other molecu
les.":? "
4950 FOR M=1 TO 100+DEM*600:NEXT M
4955 IF STRIG(0)=1 AND DEM=0 THEN GOTO
4955
4960 COLOR 0:FOR A=0 TO 72:PLOT 30,A:D
RAWTO 120,A:SOUND 0,A,10,3:NEXT A:SOUN
D 0,0,0,0
4970 IF DEM=1 THEN GOTO 15010
4980 RETURN
5000 REM ER
5002 SOUND 0,0,0,0:COLOR 3:RESTORE 501
0:PLOT 18,24:FOR A=0 TO 7:READ B,C:DRA
WTO B,C:NEXT A
5010 DATA 50,46,92,62,126,60,126,64,92
,67,50,52,20,28,18,24
5020 RESTORE 5010:PLOT 30,10:FOR A=0 T
O 7:READ B,C:DRAWTO B/1.3+16,C/1.3-8:M
EXT A
5030 RESTORE 5040:COLOR 1:FOR A=0 TO 1
5:READ B,C:PLOT B,C:PLOT B+1,C:PLOT B,
C+1:PLOT B+1,C+1:NEXT A
5040 DATA 24,32,30,30,36,42,44,40,48,5
2,50,44,64,58,62,50,76,62,76,54,96,68,
88,50,112,66,100,60,122,66,120,58
5050 RESTORE 5040:COLOR 1:FOR A=0 TO 1
5:READ B,C:B=B/1.3+16:C=C/1.3-8:PLOT B
,C:PLOT B+1,C:PLOT B,C+1
5055 PLOT B+1,C+1:NEXT A
5060 FOR A=0 TO 10 STEP -1:SOUND 0,A,
10,3:NEXT A:SOUND 0,0,0,0
5070 ? " ENDOPLASMIC RETICULUM are me
mbranes":? " which can be covered wit
h"
5080 ? " ribosomes. Proteins that are
":FOR M=1 TO 100+DEM*600:NEXT M
5085 IF STRIG(0)=1 AND DEM=0 THEN GOTO
5085
5090 ? " synthesized by the ribosomes
pass":? " into the ER and are stored
for"
5100 ? " later use. SMOOTH ENDOPLASMI
C":FOR M=1 TO 100+DEM*600:NEXT M
5105 IF STRIG(0)=1 AND DEM=0 THEN GOTO
5105
5110 ? " RETICULUM is a form of ER wi
thout":? " the ribosomes, and it serv
es to"
5120 ? " sort other molecules.":FOR M
=1 TO 100+DEM*600:NEXT M
5125 IF STRIG(0)=1 AND DEM=0 THEN GOTO
5125
5130 COLOR 0:FOR C=10 TO 70:PLOT 16,C:
SOUND 0,C,10,3:DRAWTO 126,C:NEXT C:SOUN
D 0,0,0,0
5140 IF DEM=1 THEN GOTO 15010
5150 RETURN
5200 REM CYTO
5202 SOUND 0,0,0,0:COLOR 3:FOR A=0 TO
30:B=INT(140*RN(1))+10:C=INT(65*RN(1
))+10:PLOT B,C:NEXT A
5210 COLOR 1:FOR A=0 TO 20:B=INT(140*RN
(1))+10:C=INT(65*RN(1))+10:PLOT B,C
:PLOT B+1,C:PLOT B,C+1:PLOT B+1,C+1
5215 NEXT A
5220 COLOR 2:FOR A=0 TO 10:B=INT(140*RN
(1))+10:C=INT(65*RN(1))+10:PLOT B-1
,C-1:DRAWTO B+1,C-1:PLOT B-2,C
5225 DRAWTO B+2,C:PLOT B-1,C+1:DRAWTO
B+1,C+1:PLOT B,C-2:PLOT B,C+2:NEXT A
5230 FOR A=0 TO 10 STEP -1:SOUND 0,A,
10,3:NEXT A:SOUND 0,0,0,0
5240 ? " The CYTOPLASM is the general
area":? " inside the cell which perf
orms"
5250 ? " most of the active functions
. There":FOR M=1 TO 100+DEM*600:NEXT M
5255 IF STRIG(0)=1 AND DEM=0 THEN GOTO
5255
5260 ? " are many different structure
s":? " present in the cytoplasm of"
5270 ? " different cells. Common stru
ctures":FOR M=1 TO 100+DEM*600:NEXT M
5275 IF STRIG(0)=1 AND DEM=0 THEN GOTO
5275
5280 ? " in the cytoplasm include the
":? " vacuole (a sac holding food or"
5290 ? " water), pigment and fat gran
ules.":FOR M=1 TO 100+DEM*600:NEXT M
5295 IF STRIG(0)=1 AND DEM=0 THEN GOTO
5295
5300 COLOR 0:FOR C=0 TO 75:PLOT 10,C:5
OUND 0,C,10,3:DRAWTO 148,C:NEXT C:SOUN
D 0,0,0,0
5310 IF DEM=1 THEN GOTO 15010
5320 RETURN
5400 REM MEMBRANE
5402 SOUND 0,0,0,0:COLOR 2:FOR B=0 TO
90:C=60*C05(B/1.4):PLOT B+10,(C-15)*1.
4-4:DRAWTO 10,(C-15)*1.4-4
5404 SOUND 0,(C-15)*1.4,10,3:NEXT B:50
UND 0,0,0,0
5410 COLOR 1:FOR B=0 TO 90:C=60*C05(B/
1.4):PLOT B+19,(C-15)*1.4-3:PLOT B+19,
(C-15)*1.4-4:NEXT B
5420 FOR B=0 TO 90:C=60*C05(B/1.4):PLO
T B+20,(C-15)*1.4+1:PLOT B+20,(C-15)*1
.4+2:NEXT B
5430 COLOR 3:FOR B=0 TO 90 STEP INT(4*
RN(1))+6:C=60*C05(B/1.4):PLOT B+20,(C
-15)*1.4+3:PLOT B+20,(C-15)*1.4+4
5440 PLOT B+21,(C-15)*1.4+3:PLOT B+21,
(C-15)*1.4+4:NEXT B
5450 ? " The CELL MEMBRANE is an oute
r":? " covering around the entire cel
l that"
5455 ? " is composed of polysaccharid
es. Its":FOR M=1 TO 100+DEM*600:NEXT M
5460 IF STRIG(0)=1 AND DEM=0 THEN GOTO
5460
5470 ? " primary function is to regul
ate the":? " flow of materials into a
nd out of the"
5475 ? " cell. Most particles enter t
he cell":FOR M=1 TO 100+DEM*600:NEXT M
5480 IF STRIG(0)=1 AND DEM=0 THEN GOTO
5480
5490 ? " through the membrane pores,
while":? " larger particles can be in-
gested by a"
5495 ? " process called phagocytosis.
":FOR M=1 TO 100+DEM*600:NEXT M
5500 IF STRIG(0)=1 AND DEM=0 THEN GOTO
5500
5510 COLOR 0:FOR C=0 TO 70:PLOT 10,C:D
RAWTO 120,C:SOUND 0,C,10,3:NEXT C:SOUN
D 0,0,0,0
5520 IF DEM=1 THEN GOTO 15010
5530 RETURN
7000 REM QUIT
7001 FOR A=0 TO 14:BO(A)=0:NEXT A
7002 GRAPHICS 0:SETCOLOR 2,8,0:SETCOLO
R 4,8,0:SETCOLOR 1,8,12:POKE 82,1:TRAP
7200
7010 POSITION 10,2:? " THE CELL - QUI
T":POSITION 0,3:? "
"
7020 CLOSE #3:OPEN #3,4,0,"K:"
7021 FOR A=1 TO 5
7022 RA=INT(12*RN(0))+1:IF BO(RA)=1 T
HEN NEXT A:CLOSE #3:GOTO 7110
7040 ? " NUMBER ";A

```

PITCH

R. SCHWARZ & S. OCKERS

```

10 REM *****
20 REM *      RECALL-A-PITCH      *
30 REM *      R. SCHWARZ AND S. OCKERS *
40 REM *      ATARI COMPUTER ENTHUSIASTS *
50 REM *      3662 VINE MAPLE DR.      *
60 REM *      EUGENE, OR 97405         *
70 REM *      APRIL '85 $14/YR         *
80 REM *****
90 REM
100 GOSUB 1260
110 REM *** DRAW SCREEN ***
120 ? #6;CHR$(125);POSITION 4,1: ? #6;"
PITCH RECALL";POSITION 0,6: ? #6;"SCORE
#1:";POSITION 17,6: ? #6;CHR$(4);
130 IF PLYRS=2 THEN POSITION 0,9: ? #6;
"SCORE#2:";POSITION 17,9: ? #6;CHR$(13
2):POKE 752,1
140 POSITION 0,11: ? #6;"$$$$$$$$$$$$
$$$$$$$$$$$$$$$$$$$$$";CNT1=0:CN
T2=0
150 POSITION 0,4: ? #6;"DIF #1=";E1=48;
160 IF PLYRS=2 THEN POSITION 12,4: ? #6
;"DIF #2=";E2=48;
170 GOSUB 1140:POKE 656,1:POKE 657,4: ?
"OPTION-DIFFICULTY SELECT-MPLAYERS";
180 POKE 656,2:POKE 657,7: ? "PRESS STA
RT TO BEGIN";
190 REM *** CHANGE SET-UP? ***
200 POKE CONSOLE,8
210 IF PEEK(CONSOLE)=3 THEN GOSUB 1160:
GOTO 170
220 IF PEEK(CONSOLE)=5 THEN 1090
230 IF PEEK(CONSOLE)>6 THEN 210
240 GOTO 270
250 IF RSTRT=1 THEN RSTRT=0:GOTO 120
260 REM *** PLAYER #1'S TURN ***
270 GOSUB 1140:POKE 656,1:POKE 657,13:
? "PLAYER #1 IS UP":PUP=1:E=E1:GOSUB 4
10
280 GOSUB 700:IF SCORE=1 THEN GOSUB 74
0
290 IF MIN=1 THEN GOTO 960
300 GOSUB 820:POKE 77,0:IF RSTRT=1 THE
N RSTRT=0:GOTO 120
310 IF STRIG(0)=0 OR STRIG(1)=0 THEN 3
10
320 IF PLYRS=1 THEN 250
330 REM *** PLAYER #2'S TURN ***
340 GOSUB 1140:POKE 656,1:POKE 657,13:
? "PLAYER #2 IS UP":PUP=2:E=E2:GOSUB 4
10
350 GOSUB 700:IF SCORE=1 THEN GOSUB 78
0
360 IF MIN=1 THEN GOTO 960
370 GOSUB 820:IF RSTRT=1 THEN RSTRT=0:
GOTO 120
380 IF STRIG(0)=0 OR STRIG(1)=0 THEN 3
80
390 GOTO 250
400 REM *** SELECT PITCH SUBROUTINE ***
*
410 POKE 53248,0:POKE 53249,0:POKE 532
50,0:POSITION 1,14: ? #6;"
";
420 FOR T=1 TO 300:NEXT T:POKE 656,2:P
OKE 657,11: ? "MEMORIZE THIS TONE!";
430 F=30+200*NRND(L)
440 TT=200-14*(E-48):FOR T=1 TO TT
450 IF T<TT/10 THEN SOUND 1,F,10,100*T
/TT
460 IF T>9*TT/10 THEN SOUND 1,F,10,100
*(TT-T)/TT
470 NEXT T
480 SOUND 1,0,0,0
490 GOSUB 1140:POKE 656,1:POKE 657,12:
? "WAIT ....."
500 J=1:FOR T=1 TO 500:NEXT T
510 FOR T=1 TO NRND(T)*50+20*(E-49)*4:N
EXT T
520 H=30+NRND(H)*200:SOUND 1,H,10,10
530 J=J+1:IF J<(E-49)*5 THEN GOTO 510
540 SOUND 1,0,0,0
550 GOSUB 1140:POKE 656,1:POKE 657,15:
? "MATCH THE NOTE":POKE 656,2:POKE 65
7,11: ? "THEN PRESS THE TRIGGER"
560 FOR T=1 TO 150:NEXT T
570 G=30+200*NRND(L)
580 SOUND 1,G,10,10:POKE 53248,(230-G)
*0.8+40
590 IF PUP=2 THEN 630
600 IF STICK(0)=7 THEN G=G-1
610 IF STICK(0)=11 THEN G=G+1
620 GOTO 650
630 IF STICK(1)=7 THEN G=G-1
640 IF STICK(1)=11 THEN G=G+1
650 IF G<30 THEN G=30
660 IF G>230 THEN G=230
670 IF (STRIG(0)=0 AND PUP=1) OR (STRI
G(1)=0 AND PUP=2) THEN SOUND 1,0,0,0:R
ETURN
680 GOTO 580
690 REM *** CHECK IF CLOSE ENOUGH ***
700 SCORE=0:POKE 53249,(230-F)*0.8+40:
POKE 53250,(230-F)*0.8+43:IF ABS(F-G)<
6 THEN SCORE=1
710 P$=STR$(ABS(F-G)*100/F):POSITION 1
,14: ? #6;"you are ";P$(1,3);" % off";
720 RETURN
730 REM *** UPDATE PLAYER #1 SCORE ***
740 MIN=0:FOR M=1 TO 9:POSITION CNT1,6
: ? #6;" ";FOR L=1 TO 20:NEXT L:POSITI
ON CNT1,6: ? #6;CHR$(6);:GOSUB 950
750 FOR L=1 TO 20:NEXT L:NEXT M:CNT1=C
NT1+1:IF CNT1>17 THEN MIN=1
760 RETURN
770 REM *** UPDATE PLAYER #2 SCORE ***
780 MIN=0:FOR M=1 TO 9:POSITION CNT2,9
: ? #6;" ";FOR L=1 TO 20:NEXT L:POSITI
ON CNT2,9: ? #6;CHR$(134);:GOSUB 950
790 FOR L=1 TO 20:NEXT L:NEXT M:CNT2=C
NT2+1:IF CNT2>17 THEN MIN=1
800 RETURN
810 REM *** ALTERNATE SELECTED AND RAN
DOM TONES ***
820 GOSUB 1140:POKE 656,1:POKE 657,4: ?
"THIS WAS WHAT YOU SELECTED...":POKE
656,2:POKE 657,8: ? "TRIGGER FOR NEXT";
830 SOUND 1,G,10,10:T=1
840 T=T+1:IF T>100 THEN 880
850 IF PEEK(CONSOLE)=6 THEN RSTRT=1:SOU
ND 1,0,0,0:RETURN
860 IF STRIG(0)=0 OR STRIG(1)=0 THEN S
OUND 1,0,0,0:RETURN
870 GOTO 840
880 GOSUB 1140:POKE 656,1:POKE 657,4: ?
"THIS WAS WHAT YOU HEARD...":POKE 656
,2:POKE 657,8: ? "TRIGGER FOR NEXT";
890 REM *** DING SOUND ***
900 SOUND 1,F,10,10:T=1
910 T=T+1:IF T>100 THEN 820
920 IF PEEK(CONSOLE)=6 THEN RSTRT=1:SOU
ND 1,0,0,0:RETURN
930 IF STRIG(0)=0 OR STRIG(1)=0 THEN S
OUND 1,0,0,0:RETURN
940 GOTO 910
950 FOR J=15 TO 0 STEP -1:SOUND 1,50,1
0,J:NEXT J:SOUND 1,0,0,0:RETURN
959 REM *** MIN ROUTINE ***
960 POKE 53248,0:POKE 53249,0:POKE 532
50,0
970 ? #6;CHR$(125):GOSUB 1140:POKE 656
,1:POKE 657,10: ? "PLAYER #";PUP;" MIN5
!!!":FOR M=1 TO 16
980 POSITION 1+M,6: ? #6;CG$(M,M);:GOSUB
8 950:R=INT(NRND(0)*4):GOSUB 1020
990 NEXT M:POKE 656,2:POKE 657,7: ? "PR
ESS START FOR NEW GAME";
1000 IF PEEK(CONSOLE)>6 THEN 1000
1010 GOTO 120
1020 GOSUB 1060:POSITION RX,RV: ? #6;CH
R$(38);:RETURN
1030 GOSUB 1060:POSITION RX,RV: ? #6;CH

```



```

R$(166));GOSUB 950:RETURN
1040 GOSUB 1060:POSITION RX,RY:? #6;CH
R$(6));GOSUB 950:RETURN
1050 GOSUB 1060:POSITION RX,RY:? #6;CH
R$(134));GOSUB 950:RETURN
1060 RX=INT(RND(0)*19)+1:RY=INT(RND(0)
*11)+1:IF RY=6 THEN 1060
1070 RETURN
1080 REM *** TOGGLE # OF PLAYERS ***
1090 IF PLYR5=2 THEN PLYR5=1:GOTO 1110
1100 PLYR5=2
1110 IF PEEK(CONSOL)<7 THEN 1110
1120 GOTO 120
1130 REM *** BLANK MESSAGE AREA ***
1140 POKE 656,1:POKE 657,1:? BLK$;POK
E 656,2:POKE 657,1:? BLK$;RETURN
1150 REM *** CHANGE DIFFICULTIES ***
1160 OPEN #1,4,0,"K":POKE 764,255
1170 GOSUB 1140:POKE 656,1:POKE 657,2:
? "DIFFICULTY FOR PLAYER #1 (1-9)?";
1180 GET #1,E1:IF E1<49 OR E1>57 THEN
1170
1190 POSITION 7,4:? #6;E1-48
1200 IF PLYR5=1 THEN 1240
1210 GOSUB 1140:POKE 656,1:POKE 657,2:
? "DIFFICULTY FOR PLAYER #2 (1-9)?";
1220 GET #1,E2:IF E2<49 OR E2>57 THEN
1210
1230 POSITION 19,4:? #6;E2-48
1240 CLOSE #1:RETURN
1250 REM *** SET UP PLAYER MISSILES **
*
1260 GRAPHICS 7:GRAPHICS 1:POKE 559,0:
DL=PEEK(560)+256*PEEK(561):POKE DL+11,
7:POKE DL+14,7
1270 POKE DL+16,7:POKE DL+17,7:FOR J=D
L+25 TO DL+33:POKE (J-4),PEEK(J):NEXT
J
1280 A=PEEK(106)-8:PMBASE=A*256:POKE 5
4279,A:POKE 559,46:POKE 53277,3:POKE 7
56,A
1290 FOR J=1 TO 10:POKE PMBASE+582+J,1
:POKE PMBASE+710+J,3:POKE PMBASE+838+J
,255:NEXT J
1300 REM *** COLORS ***
1310 RESTORE 1320:FOR J=0 TO 8:READ A:
POKE 704+J,A:NEXT J:POKE 623,1
1320 DATA 0,70,46,0,42,202,32,56,52
1330 REM *** MOVE 512 BYTES OF CHAR. 5
ET ***
1340 REM *** TO OPEN PM AREA ***
1350 DIM MCS$(37):RESTORE 1360:FOR J=1
TO 37:READ A:MCS$(J,J)=CHR$(A):NEXT J
:A=USR(ADR(MCS$))

```

```

1360 DATA 104,169,0,133,203,133,205,16
9,224,133,204,165,106,56,233,8,133,206
,162,2,160,0
1370 DATA 177,203,145,205,200,208,249,
230,204,230,206,202,208,242,96
1380 FOR J=PMBASE+32 TO PMBASE+39:POKE
J,255:NEXT J
1390 RESTORE 1400:FOR J=PMBASE+48 TO P
MBASE+55:READ A:POKE J,A:NEXT J
1400 DATA 15,12,12,124,252,252,120,0
1410 DIM BLK$(38):BLK$(1)=" ":BLK$(38)
=BLK$:BLK$(2)=BLK$:DIM CG$(16):CG$="Co
lor Attributions!"
1420 DIM D$(1),P$(4):PLYR5=2:CONSOL=53
279:E1=49:E2=49:R5TRT=0
1430 POKE 559,46:RETURN

```

HAPPY SLOWDOWN

FROM LAST MONTH

```

10 REM ----Shane Rolin Presents:
20 REM Happy Drives BASIC slowdown
30 REM program.
40 REM ----
50 REM program by SHANE ROLIN
60 REM of Pittsburgh A. C. E.
70 REM
90 CLR :TRAP 730
100 DIM DAT$(70),DSKINV$(3),SIOV$(3)
110 REM
120 REM Let's define Page Three
130 REM
140 DUNIT=769
150 DCOMM=770
160 DSTATS=771
170 DBUFLO=772
180 DBUFHI=773
190 DAUX1=778

```

```

200 DAUX2=779
210 REM
220 REM Let's define ML subroutines
230 REM
240 DSKINV$(1)=CHR$(32)
250 DSKINV$(2)=CHR$(83)
260 DSKINV$(3)=CHR$(228)
270 REM
280 SIOV$(1)=CHR$(32)
290 SIOV$(2)=CHR$(89)
300 SIOV$(3)=CHR$(228)
310 REM
320 REM Let's define slow down string
330 REM
340 FOR LOOP=1 TO 51
350 READ BYTE
360 DAT$(LOOP)=CHR$(BYTE)
370 NEXT LOOP
380 REM
390 DATA 38,17,52,20,21,87,55,133
400 DATA 134,76,55,31,76,33,8,76
410 DATA 38,25,76,54,29,76,195,29
420 DATA 76,195,29,76,195,29,76,50
430 DATA 31,169,1,141,235,1,169,29
440 DATA 141,14,8,169,195,141,13,8
450 DATA 76,65,24
460 REM
470 REM Let's find addr. of SIOV$
480 REM
490 SIOVADDR=ADR(SIOV$)
500 REM
510 REM Let's find addr. of DSKINV$
520 REM
530 DSKADDR=ADR(DSKINV$)
540 REM
550 REM Let's find Lo/Hi of DAT$
560 REM
570 ADDR=ADR(DAT$)
580 DATHI=INT(ADDR/256)
590 DATLO=ADDR-(DATHI*256)
600 REM
610 REM Let's run the program!!
620 REM
630 POKE DUNIT,2
640 POKE DCOMM,87
650 POKE DAUX1,0
660 POKE DBUFLO,DATLO
670 POKE DBUFHI,DATHI
680 POKE DAUX2,8
690 X=USR(DSKADDR)
700 POKE DCOMM,81
710 POKE DSTATS,0
720 X=USR(SIOVADDR)
730 END

```

THE CELL CONT

```

7050 RESTORE 11970+RA*30:READ DA$: ? DA
$: ? :READ AN$:READ DA$: ? DA$:READ DA$:
? DA$
7060 B0(RA)=1: ? "YOUR ANSWER ";:GET
#3,AN: ? CHR$(AN)
7068 IF CHR$(AN+32)<AN$ THEN GOTO 708
0
7070 RI=RI+1: ? "THAT IS RIGHT!"
7074 FOR D=80 TO 10 STEP -10:SOUND 0,D
+INT(10*RNDD(1)),10,10:SETCOLOR 2,0,0:F
OR M=1 TO 10:NEXT M
7078 SETCOLOR 2,0,15:NEXT D:SETCOLOR 2
,0,0:GOTO 7090
7080 ? : ? "THAT IS WRONG, THE ANSWER I
S ";AN$
7084 FOR D=50 TO 150 STEP 10:SOUND 0,D
+INT(10*RNDD(1)),10,5:FOR M=1 TO 8:NEXT
M:NEXT D
7090 SOUND 0,0,0,0: ? : ? "
_____": ? :NEXT A:CL
OSE #3
7110 IF RI<3 THEN GOSUB 8000:GOTO 40
7120 IF RI<4 THEN GOSUB 9000:GOTO 40
7130 IF RI>4 THEN GOSUB 10000:GOTO 40
8000 GRAPHICS 7:SETCOLOR 4,0,0:SETCOLO
R 2,0,0:SETCOLOR 0,3,0:COLOR 1
8010 FOR A=1 TO 80:B=INT(150*RNDD(1))+5
:C=INT(70*RNDD(1))+5:PLOT 80,40:DRAMTO
B,C:SOUND 0,A*2,0,8
8020 NEXT A:SOUND 0,0,0,0: ? " THAT IS
TERRIBLE!": ? "YOU MADE ONLY ";RI;" COR
RECT ANSWERS!"
8030 FOR A=8 TO 0 STEP -1:FOR M=1 TO 5
0:NEXT M:SETCOLOR 0,3,A:NEXT A:SETCOLO
R 0,0,0:RETURN
9000 ? : ? "THAT WAS OKAY.": ? "YOU MADE
";RI;" CORRECT ANSWERS.":FOR M=1 TO 1
000:NEXT M:RETURN
10000 GRAPHICS 2+16:SETCOLOR 4,0,0:SET
COLOR 0,3,0:SETCOLOR 1,0,0:POSITION 2,
2: ? #6;"CONGRADULATIONS!"
10005 POSITION 2,4: ? #6;"you answered"
:POSITION 2,5: ? #6;"all questions":POS
ITION 4,6: ? #6;"CORRECTLY!"
10010 FOR A=20 TO 0 STEP -2:FOR B=20+A
TO 5 STEP -3:SOUND 0,8,10,10:SETCOLOR
4,0,INT((85-B)/5.3)
10020 NEXT B:NEXT A:SETCOLOR 4,0,0:FOR
B=5 TO 40 STEP -1:SOUND 0,8,10,10:NEX
T B:SOUND 0,0,0,0:FOR M=1 TO 300:NEXT
M:RETURN
11999 REM TEXT DATA FOR QUIZ
12000 DATA What structure controls the
cell?,c

```

12010 DATA a. vacuoles	c. nuc	of polysaccharides?,b
leus		12310 DATA a. DNA
12020 DATA b. DNA	d. cyt	leus
oplasm		12320 DATA b. cell membrane
12030 DATA Where is the ribonucleic ac		d. Mit
id?,b		ochondria
12040 DATA a. cell membrane	c. lys	12330 DATA How are large particles ing
osomes		ested?,d
12050 DATA b. nucleoli	d. rib	12340 DATA a. replication
osomes		c. exo
12060 DATA How many centrioles are in		cytosis
all animal cells?,b		12350 DATA b. numeration
12070 DATA a. 50-100	c. 5	d. pha
12080 DATA b. 2	d. non	gocytosis
e		
12090 DATA What does the centriole con		
trol?,b		
12100 DATA a. golgi bodies	c. nuc	
leus		
12110 DATA b. mitotic spindle	d. nuc	
leoli		
12120 DATA Where is the mitochondria?,		
a		
12130 DATA a. cytoplasm	c. nuc	
leus		
12140 DATA b. cell membrane	d. cil	
ia		
12150 DATA What molecule is formed in		
the mitochondria?,a		
12160 DATA a. DNA	c. ATP	
12170 DATA b. glucose	d. RNA	
12180 DATA What has the role of comple		
xing proteins?,c		
12190 DATA a. ER	c. gol	
gi body		
12200 DATA b. ribosomes	d. DNA	
12210 DATA Where are proteins stored?,		
b		
12220 DATA a. cytoplasm	c. nuc	
leus		
12230 DATA b. ER	d. Mit	
ochondria		
12240 DATA Which structure has attache		
d ribosomes?,a		
12250 DATA a. ER	c. DNA	
12260 DATA b. vacuole	d. lys	
osomes		
12270 DATA Where are most of the funct		
ions of the cell performed?,c		
12280 DATA a. ATP	c. cyt	
oplasm		
12290 DATA b. nucleus	d. rib	
osomes		
12300 DATA What is primarily composed		

APRIL
MEETING
WED. THE 10TH
SOUTH EUGENE
HIGH
7:30PM

BRING YOUR FRIENDS



Ralph Walden Teaches

Assembly Language #5

A common way of initializing a string array is to read it via DATA statements. This has the advantage that one can print it in ACE and other newsletters, but it tends to initialize rather slowly compared to a STRING\$ = "text". This program will set up the string assignments for you. To use it, merge the program with your own program. Let's say you have the following program segment:

```
10 DIM STRING$(20):RESORE 100:FOR I=1 TO 20:READ A:STRING$(I)=CHR$(A):NEXT I
```

What you will do now is add the line and run the program.

```
11 GOSUB 32000
```

The screen will clear, and you will be asked the name of the string, then the line number you want the string to start initializing on, and the increment between lines. You will then see the lines being created on your screen, and the READY prompt when it's done. You can now delete lines 10,11 and all of the respective DATA statements. On paper, this probably sounds pretty complex, but in practice it isn't. You just need to decide ahead of time where you want the lines of initialization, then add a GOSUB 32000 after the string you want changed is initialized via DATA statements, and run the program. (Note: if you are using BASIC XL do NOT use FAST model). When you are done using the program, you can do a GOSUB 32000, and when it asks for the string name, just press RETURN, and the program will remove itself from your main program. Notice that line 32109 will first SAVE the program. The reason for this is that Atari Basic, and occasionally even BASIC XL, will crash when deleting lines. This will give you a back-up should that occur.

Some printing errors in the previous programs:

#3 (Feb): on lines 170 and 410, the cross should have been a +

#4 (March): on line 310, should be @TEXT —

Ralph Walden

JOYSTICK DRIVER DEMO

```
10 REM*****
*
11 REM*
*
12 REM* LOGICAL OPERATION ROUTINE
*
13 REM* by M.J. Cunningham 1/7/85
*
14 REM* Starfleet ATARI User Group
*
15 REM*
*
16 REM*****
*
17 REM
18 REM The following are 16 bit machine
  language routines allowing true AND,
  OR, and EXCLUSIVE OR logical
19 REM operations. Unlike the ATARI
  BASIC logical operations returning o
  nly a 1 or 0 value,
20 REM these return the full 16 bit
  result of 16 bit integer logical
  operations.
21 REM
22 REM Set up by a GOSUB to the
  appropriate IMIT section. Call as
  shown in the examples.
23 REM
355 REM Init for logical exclusive or
356 REM Example of use:
357 REM RESULT = USR(ADR(LEOR$),ARG1,ARG2)
358 DIM LEOR$(21):FOR I=1 TO 21:READ A:LEOR$(I,I)=CHR$(A):NEXT I:STOP
359 DATA 104,32,68,218,104,133,203,104,133,204,104,69,203,133,213,104,69,204,133,212,96
375 REM Init for logical or
376 REM Example of use
377 REM RESULT = USR(ADR(LOR$),ARG1,ARG2)
378 DIM LOR$(21):FOR I=1 TO 21:READ A:LOR$(I,I)=CHR$(A):NEXT I:RETURN
379 DATA 104,32,68,218,104,133,203,104,133,204,104,5,203,133,213,104,5,204,133,212,96
395 REM Init for logical and
396 REM Example of use
397 REM RESULT = USR(ADR(LAND$),ARG1,ARG2)
400 DIM LAND$(21):FOR I=1 TO 21:READ A:LAND$(I,I)=CHR$(A):NEXT I:RETURN
401 DATA 104,32,68,218,104,133,203,104,133,204,104,37,203,133,213,104,37,204,133,212,96
```

```
100 REM *****
101 REM * Joystick Driver Demo
102 REM * by Jonathan Buckheit
103 REM *****
104 REM
105 REM - Joystick Driver -
106 REM
107 JOYSTICK=ADR("joystick")
108 REM
109 REM - Set up Screen -
110 REM
111 GRAPHICS 0:POKE 710,0:POKE 752,1:
112 REM
113 REM - Set up Delta Change Array -
114 REM
115 DIM DELTA(7,1):FOR JB=0 TO 7:READ DELTA(JB,1):DELTA(JB,0)=DELTA(JB,1):DELTA(JB,1)=DELTA(JB,0):NEXT JB
116 DATA 0,-1,1,-1,1,0,1,1,0,1,-1,1,-1,0,-1,-1
117 REM
118 REM - Put Character onto Screen -
119 REM
120 POSITION XPOS,YPOS: "0";
121 REM
122 REM - Read Joystick -
123 REM
124 JB=USR(JOYSTICK,0):IF JB=8 THEN 340
125 REM
126 REM - Check Coordinates -
127 REM
128 DELTA=XB*DELTA(JB,0):DELTA=YB*DELTA(JB,1):IF DELTA<1 OR DELTA>380 OR DELTA<-1 OR DELTA>22 THEN 340
129 REM
130 REM - Erase Old Character -
131 REM - Update Coordinates -
132 REM
133 POSITION XPOS,YPOS: " ":XPOS=DELTA:YPOS=DELTA:GOTO 300
```

```
32000 GRAPHICS 0: "KSTRING NAME (RET. TO ERASE):":DIM QZ$(20):INPUT QZ$:IF QZ$="" THEN POKE 842,13:GOTO 32019
32001 ? "KLN.N,INC.":INPUT LN,IN: "K+L=LEN("QZ$"):DIM THE$(L):THE$="":QZ$="":CONT":POSITION 0,0:POKE 842,13:STOP
32002 Z=0:DIM CLEAR$(L):CLEAR$=THE$:FOR N=1 TO L:IF THE$(N,N)=CHR$(34) OR THE$(N,N)=CHR$(155) THEN THE$(N,N)="A"
32003 NEXT N:N=1:L=L+80
32004 L=L-80:IF L<80 THEN Z=L:GOTO 32006
32005 Z=Z+80
32006 ? "K+":? LN," ":QZ$,"$("N;N)";CHR$(34):POKE 766,1: THE$(N,Z): "CONT":POSITION 0,0:STOP
32007 N=N+80:POKE 766,0:LN=LN+IN:IF L>79 THEN 32004
32008 Z=1:FOR N=1 TO L:IF CLEAR$(N,N)=CHR$(34) OR CLEAR$(N,N)=CHR$(155) THEN GOTO 32010
32009 NEXT N:GOTO 32017
32010 IF Z>1 THEN 32013
32011 ? "K+":? LN," ":QZ$,"$("N;N)";N;N)=CHR$(ASC(CLEAR$(N,N)):"");
32012 Z=Z+1:GOTO 32009
32013 Z=Z+1:IF Z>5 THEN 32015
32014 ? " ":QZ$,"$("N;N)";N;N)=CHR$(ASC(CLEAR$(N,N)):"");GOTO 32009
32015 Z=1: "CONT":POSITION 0,0:STOP
32016 LN=LN+IN:GOTO 32011
32017 ? "CONT":POSITION 0,0:STOP
32018 POKE 842,12:GRAPHICS 0:END
32019 SAVE "D1:TMP": "K+": "32000": "32001": "CONT":POSITION 0,0:STOP
32020 ? "K+":FOR N=32002 TO 32021: "NEXT N: "POKE 842,12:GR.0:END":
32021 POSITION 0,0:STOP
```

DISK LISTING PROGRAM

BY DICK KUSHNER

```

10 REM *****
20 REM * DISK LISTING PROGRAM by *
30 REM *DICK KUSHNER & GORDAN BANKS*
70 REM *****
80 REM * JACQ NEWSLETTER - Dec.1984 *
90 REM * $20/Year *
91 REM *Contact:Ron Kordos *
92 REM * 201 Lake Valley Rd *
93 REM * Morristown, NJ 07960*
95 REM *****
100 DIM B$(17),N$(40),F$(900),S$(16),F
B$(12000):X=1:O=X-K:P=7:FB$="":POKE 82
,2:GOSUB 470
110 GOSUB 350: "1. Load Printer with
Labels or paper."
120 ? "42. Turn on printer and interfa
ce."
130 ? "43. Insert diskette into Drive
1."
140 POKE 752,X: "44Press RETURN when
ready";:INPUT N$:?
150 TRAP 150:GOSUB 350: "PRINT HOW MA
NY COLUMNS?": "45Select 2 to 3 for 3.
5 in. labels."
160 ? "46Select 2 to 4 for 5 in. labels
.": "47Select 1 to 6 columns for paper
."
170 ? "48HOW MANY COLUMNS (1 to 6)";:I
NPUT COL:IF COL<X OR COL>6 THEN 150
180 GOSUB 350: " Do you want the fi
les printed?": "49alphabetically (Y/
N)";:INPUT N$:IF N$="" THEN 180
190 IF N$(X,X)="" THEN SORT=X:GOTO 21
0
200 IF N$(X,X)<>"N" THEN 180
210 TRAP 310:OPEN #P,0,"P": "50";:Ee
E3-50"
220 CLOSE #2:GOSUB 350: "Enter 'END'
to quit.":IF LEN(FB$) THEN ? "4Enter '
SORT FILE' to print all files."
230 ? "4Enter a diskname for next disk
label.4":INPUT N$:IF N$="END" THEN GR
APHICS 0:END
240 IF N$="SORT FILE" THEN POKE 205,12
:POKE 204,12:GB=USR(1568,ADR(FB$),LEN(
FB$)/12):GOTO 360
250 OPEN #2,0,"D":*,:F$=""
260 INPUT #2,B$:IF LEN(B$)<17 THEN GOT
O 330
270 F$(LEN(F$)+X)=B$(3,10):F$(LEN(F$)+
X)="":F$(LEN(F$)+X)=B$(11):FB$(LEN(FB
$)+1)=F$(LEN(F$)-15,LEN(F$)-4):GOTO 26
0
280 GOSUB 350:POKE 752,X: "51Printing.

```

```

..";N$:L=X:CP=0:FOR I=X TO LEN(F$) STE
P 16
290 ? #P:F$(I,I+15); " ";:CP=CP+X:IF
CP=COL THEN ? #P:CP=0:L=L+X:IF COL>1
THEN IF L=10 THEN L=0: #P: #P
300 NEXT I:FOR I=L TO 11: #P:NEXT I:G
OTO 220
310 TRAP 40000:IF PEEK(195)=138 THEN 1
10
320 ? :? "ERROR ";PEEK(195); " AT LINE
";PEEK(186)+256*PEEK(187):STOP
330 IF SORT THEN POKE 205,16:POKE 204,
16:GB=USR(1568,ADR(F$),LEN(F$)/16)
340 FOR I=X TO (18*COL-(21+LEN(N$)))/2
: #P: " ";:NEXT I: #P:N$; " HAS ";B$:G
OTO 280
350 GRAPHICS 0: "52 DISK LISTING PR
OGRAM";:FOR I=0 TO 15 STEP 0.5:SOUN
D 0,20,10,15-I:NEXT I:RETURN
360 GOSUB 350: "Ready to print total
directory.": "53Insure that printer is
loaded with 4PAPER, NOT LABELS"
370 ? "4Enter '1' for tiny print.":? "
4 '2' for small print.":? "4
'3' for normal print."
380 ? "4 '4' for LARGE print."
390 ? "44Enter choice when ready";:IMP
UT N$:IF N$="1" THEN ? #P:"E3-50":?
#P:GOTO 440
400 IF N$="2" THEN ? #P:"Ee,EN":? #P:
GOTO 440
410 IF N$="3" THEN ? #P:"EeEN":? #P:G
OTO 440
420 IF N$="4" THEN ? #P:"EeENEN":? #
P:GOTO 440
430 GOTO 360
440 L=X:FOR I=X TO LEN(FB$) STEP 12:
#P:L; " ";:IF L<10 THEN ? #P: " ";
445 ? #P:FB$(I,I+11):L=L+X:NEXT I:GOSU
B 350
450 ? "Enter '1' for another copy":? "
4RETURN to End program";:INPUT N$:IF N
$="1" THEN 360
460 GRAPHICS 0:END
470 IF PEEK(1568)<>104 THEN GOSUB 350:
? "One moment please...":FOR I=1568 T
O 1693:READ A:POKE I,A:NEXT I
480 POKE 203,0:POKE 206,0:RETURN
490 DATA 104,104,133,217,104,133,216,1
04,133,209,104,133,208,169,0,133,218,1
33,207,162,1,165,216,133,214,165,217
510 DATA 133,215,24,165,214,133,212,10
1,205,133,214,165,215,133,213,105,0,13
3,215,164,203,165,206,240,10,177,214

```

```

520 DATA 209,212,144,44,240,12,176,19,
177,214,209,212,144,13,240,2,176,30,20
0,196,204,240,227,176,23,144,223,169
530 DATA 1,133,218,164,205,136,177,214
,72,177,212,145,214,104,145,212,192,0,
208,241,232,224,0,208,2,230,207,228
540 DATA 208,208,172,165,209,197,207,2
08,166,165,218,201,0,208,144,96

```

SURF SOUND

```

10 REM ***REAL SURF SOUND***
20 REM ***BY LEE MINARD***
30 REM ***STARFLEET, DENVER***
40 GRAPHICS 2:SETCOLOR 4,8,6:SETCOLOR
2,0,8
50 ? #6: "54 #6: #6: #6: #6: SURF'S UP
!"
60 POKE 709,0:POKE 752,1
70 ? " Real Surf Sound"
80 ? " by Lee Minard"
100 X=9
200 FOR J=50 TO 11 STEP -1
230 SOUND 3,225,4,X:SOUND 2,0,8,X
234 X=X-0.5
235 IF X=0 THEN X=1
236 IF J=40 THEN X=7
237 IF J=19 THEN X=7
238 IF J=35 THEN X=5
239 IF J=1 THEN X=11
240 SOUND 0,J,8,4:SOUND 1,J+5,8,3
250 FOR WAVE=1 TO 20+RND(0)*100:NEXT W
AVE
265 NEXT J
300 FOR J=10 TO 0 STEP -0.5
330 SOUND 3,225,4,X:SOUND 2,0,8,X
334 X=X-0.5
335 IF X=0 THEN X=1
336 IF J=5 THEN X=6
337 IF J=1 THEN X=9
340 SOUND 0,J,8,4:SOUND 1,J+5,8,3
350 FOR WAVE=1 TO 20+RND(0)*150:NEXT W
AVE
365 NEXT J
400 FOR DLAY=1 TO 300+RND(0)*300:NEXT
DLAY
500 GOTO 100

```

DOUBLE SYNFILE

(reprint: JAGC, Feb. 1985)

Want to operate on SynFile+ records in double density? Joe Zarnitz of JAGC provides an undocumented way to do it! — JB, editor

First you should make a copy of the file using the "copy" function of SynFile+. This procedure "empties the trashcan" and on files where records have been deleted will shorten the number of bytes considerably.

The next step is to change density using the "copy file" function of your favorite DOS with one drive set up for each density. [Some drives will handle this operation in a single-drive configuration]. Be sure to get all 4 files (suffixs *.IDX, *.TBL, *.D01, and *.CNF). The wildcard suffix will catch them all.

Now boot up SynFile+ and "OPEN" your file in its newest version. The density should adjust automatically. Select your file and then go to the "RECORDS" option, select "REINDEX" and reindex the file. You can reindex by the same criteria as before or by a new one, but you must go through this procedure if you want this to work.

Now you can use your file in any way you wish, just as though you had made the original file in double density form. If you have any questions, call me at 201-539-4212.

— Joe Zarnitz

USERCOMP

(Reprint: U.K. Atari User's Club)

UserComp is a simple (?) single-pass machine-language assembler which takes most of the hassle out of creating User subroutines. It is designed to create a string directly, eliminating messy data reads (though this facility is still available). Although no labels (user-defined names for certain line numbers) are allowed, it is very easy and efficient to use, and is designed to save many of the keystrokes demanded by other compilers.

Loading

If your program is not yet written, CLOAD the Compiler. The program loads to 9000, whereupon you can create the USRs required, eliminate the Compiler by its built-in utility, and continue to write the rest of your program. If your program is already partially written, then procede as above, and LIST the USRs to tape, then ENTER them into your main program.

Using the Program

On running the program, a "form" will be displayed. The first op-code "PLA" is also displayed, both as an example of the format in which to enter the codes, and because all USRs will need this sooner or later. (It can be eliminated if desired by use of the Directive RUB). We recommend you first jot down your program on paper, rather than compose it on the screen, as you must know in advance to which line numbers you are branching.

The Operation Code Field

The solid question mark invites the entering of an alphabet character to form a 3-letter op-code. Some keys are locked out, and no editing is possible. If a mistake is made, enter three letters anyway and let the program reject it as invalid, or use the Directive RUB on the next line to erase it. No space or return is required; the program takes-off after the third keystroke to decide whether an address is required.

The Operand

If the program decides the op-code mnemonic is a valid one requiring an address field, then the cursor will invite one. For the first keystroke, valid responses are:

A for the Accumulator (shift/rotate group only)

Hash or Equals to indicate the "Load Immediate" mode

Dollar to prepare for Hexadecimal input

Open Bracket for Indirect Page zero address modes

Digits to commence a Page zero or Absolute address

T to mean a location in Page zero called "Temp" for general usage. (The idea is to avoid having to remember which page zero locations are available to the user, saving keystrokes, and having a recognizable label on the screen — the compiler expands the label on-screen from "T" to "TEMP").

H, L to mean the locations in page zero where the "result" (if used) is passed back to Basic. H and L refer to the high and low bytes of this result, interpreted by Basic as a real positive integer from zero to 65536. If not used, Basic will assign a "result" to the USR call based on whatever number was last computed, for example the address of the USR string.

D to mean a pair of locations in page zero to be used as a destination pointer for (indirect) Y operations.

S to mean another pair used as a "source pointer". (Copying character sets or moving data requires 2 pointers). To access the high byte of these pairs, use the PLUS symbol after the label.

I to mean "indirect via" one of these pointers. Which pointer is used depends on whether a Load or Store instruction is being called.

M to mean the location MEMTOP (106) which Basic is forbidden from encroaching beyond.

Second and subsequent valid keys are:

Digits and letters (A to F), Indexes (X,Y), Comma and Bracket.

A Plus sign may be used to add 1 to any address — used to access the high location of pairs "D" and "S".

USERCOMP	INSTRUCTION	SET
OP. Operation	Modes	
ADC Add with Carry	8	
ADD MACRO (CLC, ADC)	8	
AND Logical AND to A	8	
ASL Arithmetic Shift Left	5	
BCC Branch on Carry Clear	1	
BCS Branch on Carry Set	1	
BEQ Branch on Equal (Z flag set)	1	
BIT Test Memory bits (flags N,V,Z)	2	
BMI Branch if Minus (N flag set)	1	
BNE Branch if Not Equal (Z reset)	1	
BRK Break Assembly (NOT 6502 BRK)	-	
BPL Branch if Plus (N reset)	1	
BVC Branch if overflow clear	1	
BVS Branch if overflow (bit 6 to 7)	1	
BYT Inpute Byte or Word (Note 5)	2	
CIO Macro (JSR Central I/O)	-	
CLC Clear Carry flag	-	
CLD Clear Decimal arith. flag	-	
CLV Clear overflow flag	-	
CMP Compare to A	8	
CPX Compare to X	3	
CPY Compare to Y	3	
DCP Decrement&compare to A (Note2)	4	
DEC Decrement memory	3	
DEX Decrement index X	-	
DEY Decrement index Y	-	
END Compile & Print USR	-	
EOR Exclusive OR to A	8	
INC Increment memory	3	
INX Increment X	-	
INY Increment Y	-	
LAX Load A & X (Note 2)	3	
LDA Load Accumulator	8	
LDX Load Index X	3	
LDY Load Index Y	4	
LSR Logical Shift Right	5	
NEW (See Note 3)	-	
NOP No Operation	-	
ORA Logical OR to A	8	
PHA Push A onto stack	-	
PHP Push Processor status	-	
PLA Pull A from stack	-	
PLP Pull status flags	-	
PUL Macro (PLA, STA)	7	
PUT Macro (LDA, STA) (Note 4)	X	
RES Macro (NOP,NOP,NOP)	-	
ROL Rotate left through carry	5	
ROR Rotate right through carry	5	
RTS Return from USR to Basic	-	
RUB Delete last line entered	-	
SBC Subtract with Carry	8	
SEC Set Carry flag	-	
SED Set Decimal mode	-	
STA Store Accumulator to Memory	7	
STX Store Index X to Memory	4	
STY Store Index Y	4	
SUB Macro (SEC,SBC)	8	
TAX Transfer A to X	-	
TAY Transfer A to Y	-	
TSX Transfer Stack pointer to X	-	
TXA Transfer X to A	-	
TXS Transfer X to Stack Pointer	-	
TYA Transfer Y to A	-	

Note 1. Macros here simply mean pre-defined instructions.

Note 2. See "Extra Instructions", COMPUTE!, October, 1983.

Note 3. Same as RESET, GOTO 9000. Do NOT use after Reset as Basic will wipe out the program!

Note 4. Use as the equivalent of POKE, except the data comes before the address, with LDA (immediate), STA page zero or absolute. The equivalent of a MOVE when used as LDA (location), STA (location). The only restriction on the number of addressing modes comes with fitting it onto the screen.

Note 5. Use as data and to support those op-codes not mentioned here. For example, JSR can be input as BYT 32, BYT (address word). When the number of addressing modes is specified (in the MODES column) as anything other than "." then an address field is expected of you. For Branches, only the Line Number is acceptable. A Hash sign, or an Equals sign, are both acceptable as an "immediate" command.

A Dollar sign sets the input to Hex mode (default decimal).

The address field is terminated by a Space, a Return, or sometimes by the indexing symbols X, Y, or).

Atari File Developer

This past summer I worked on an Apple computer using a program called Appleworks. (This is probably the best microcomputer program I have ever seen). In using Appleworks, I encountered many scrolling disk directories from which one could pick his choice (or choices) and then have the program operate on them. I then took it upon myself to write a similar thing for Atari using my then newly acquired Action! cartridge. After many hours of blood, sweat and tears I came up with the scrolling directory routines and a program which I affectionately call 'FID' (after an Apple program which does many of the same functions).

This program will allow you to do mass file locks, unlocks, deletes, and copies, in addition to basic disk formatting. Mass file copying is done by first telling the program what files you want copied. Then the program reads them all in from the source disk into a memory buffer, and writes them back out to the destination disk, with only a single disk swap. Unfortunately for you multi-drive system owners, FID was written for only a single drive system, i.e. it will only operate on the disk in drive 1. Actually, if you have two drives, the main function of FID (mass file copies with few disk swaps) is no big deal anyway.

While all the functions of FID are available from your DUP menu, FID lets you do these in an efficient, user-friendly manner. The only keys used are the up and down arrows, numbers 1 through 5, RETURN, and the START button. I have given this program to a neighbour boy with an Atari who couldn't figure out the DUP menu, and now he uses FID all the time.

USING THE PROGRAM:

When you load the FID program, you will be faced with a menu of 5 items. Before you choose the function you want executed, you will have to place the disk you want operated on in drive one. Then push the number, and the program will read in the directory and show you a portion of it. To select files to be operated on, press the RETURN key. This will cause the file pointed to by the pointer to be displayed in inverse video, indicating its selection. To move the pointer up and down, simply press the + and * keys (without holding down control). To move the pointer quickly, hold down control and press these keys. To deselect any files you may have selected by accident, simply position the pointer across from them and press RETURN again. When you are finished making selections, press the START button, and the files will be operated on.

If you just wish to see the directory on a disk, choose one of the options Lock, Delete, or Unlock, and then don't press RETURN. When you are done viewing the menu, press START.

To use the format option, press the 5 button. The program will attempt to check if the disk is already formatted, if it is you will be asked to confirm your choice. If it isn't, then it will be automatically formatted. NOTE: It sometimes takes quite a while to decide the disk is not formatted. If this delay is unacceptable, you might want to modify the procedure Checkformat() so it only says Format("D1:"). This will mean removing the rest of the body of this procedure.

COPYING FILES:

FID was originally written for this and only this purpose. I had gotten angry at DOS for making me switch disks so much when I was copying a bunch of small files. What FID will try to do is read as many of the selected files into memory as it can, then dump these all out onto the destination disk. Generally speaking, it works well, but for very large files the DUP copy should be used — FID gets confused if it can't read a very large file into its memory buffer at a single pass. Also, avoid copying DOS files with it — better use DUP to write DOS.

I have used FID in conjunction with RANA SmartDOS to transfer a number of single density music files to double density. With only a single drive, density conversion can be a very long, boring sequence of disk swaps. With FID and SmartDOS, the disk swaps are cut down drastically (if small files are being copied). To use FID in this capacity, simply boot your system with RANA SmartDOS and then either Load FID or boot it from the ACTION! monitor.

COMPILING:

When you compile this program, some special precautions must be taken. First of all, you should type the program into your editor and compile it while it is still in memory until all the syntax errors are removed. Then save the text file to disk from the ACTION! editor. Clear the editor, and enter the monitor. Type C "FID" (or whatever name you used) and wait while it compiles. Now type ?SE from the monitor. Make a note of the last number printed out by the monitor, and make sure "codetop" is at least 100 greater than this value. If it is, then type w "FID.COM" to write out an executable file. If codetop isn't large enough, go to the editor, load the file in, change it, saved the file, and recompile. Note that this COM file will need the ACTION! cartridge to be present to run. If you have the Runtime package, you may want to compile and save a runtime version of FID so you will be able to run it without the cartridge in. In this way, you gain 8K of buffer space.

THE PROGRAM:

I am embarrassed to say that this program is not a very good example of an ACTION program. I wrote it during the summer of 1984 (almost a year ago!!) before I had taken any university and by my prof's standards this program is a mess. However, for those who want to spend some time, the procedures from Read Directory down to Execute could be copied into another program, and with the right global variable declarations, the scrolling directory should be able to be used with a minimum of effort. The execute procedure would have to be called from the program, and would get the filenames selected and presumably either operate on them or return them to the main program. Let me know if you get these going in your own programs, or if any suggestions for features in FID that you would like to see added. My address is:

Dale Lutz
3504-104 St.
Edmonton, AB T6J 2J7, Canada

THE CELL

THE CELL is an educational program using graphics to give the user a colorful tour of a typical animal cell. It may be used by anyone who wishes to learn a bit about cells (and it is a great review for a biology test!). I spent quite a bit of time to write and debug it, and it can be very educational.

The program displays a large color image of a single cell, and a blinking cursor. The cursor may be moved, using the joystick, to any of eight different parts of the cell. When you wish to view a part of the cell in greater detail, press the trigger and a magnified image will appear, along with some textual information. When you feel you have absorbed everything, press START and a multiple-choice quiz will commence.

THE CELL was originally written for a science-fair project a year ago. When I originally wrote THE CELL, I did not understand such "complex" things as Player/Missiles or Page Flipping, so the techniques used in the program are simple. Each part of the cell is drawn in graphics mode 7, by routines using normal PLOT and DRAWTO commands. The multiple choice quiz uses questions and answers stored in DATA statements beginning at line 12000, and more questions could be added. The major routines of THE CELL are identified with REM statements, so it should be easy to modify the program if you wish.

THE CELL requires at least 32K to operate. It might take quite a bit of typing to enter the whole program, but it should be worth it!

—Paul Freeman

ATARI ANIMATE

ANIMATION - USING CHARACTER GRAPHICS

Using any one of the available drawing programs such as DataSoft's Micro-Painter, it is possible to design very detailed and colorful screens. These programs use ANTIC mode 14 (sometimes known as GRAPHICS 7.5), which provides 4 colors with a resolution of 320x192 pixels. The only disadvantage is the mode uses almost 8K of screen memory. If you wish to use your ANTIC 14 designs in your own programs, this memory limitation can cause problems.

To solve this problem, Jerry White has created an excellent graphics package called **Atari Animate**. The program allows you to load in a screen created with Micro-Painter (or other programs), and convert the screen into a font for use with ANTIC mode 4. ANTIC 4 is a multi-color character mode which has the same resolution and number of colors as ANTIC 14, but ANTIC 4 uses less than 1K of memory - a big savings! **Atari Animate** consists of a disk filled with many programs all designed to work together. The main programs of the package are:

S2FONT - The main utility program which converts graphics screens into character fonts.

ANIMATE - A demonstration program which displays a walking robot. Once you have seen this demonstration, you will realize the advantages of using character graphics.

DISPDISK - A utility which displays on the screen a font created by S2FONT.

FONT2STR, SMP2STR, DISPSTR, SCR2STR - A series of utilities which convert character fonts and displays into BASIC strings.

STR2SMEM.SRC - An Assembler source code for a string display routine used by the ANIMATE program.

The documentation to the programs is very complete (17 pages), and it serves as a tutorial on how to use character graphics for animation purposes. The author, Jerry White, shows you every step of the process needed to convert your graphics screens into a functioning part of your program. **Atari Animate** is an excellent addition to anyone's library of Atari software, and I plan to use it extensively. It is available for \$12.00 (very reasonable!) from: Jerry White, 18 Hickory Lane, Levittown, NY 11756.

— Paul Freeman

PQ (\$45, Suncomm, 1985 Design: Tom Quinn Jr., Programming: Joe Hellesen), or Party Quiz, is a basic computer trivia game. PQ allows up to four players to play in a fast pace question and answer session. The package also includes hardware: 4 game controllers and adaptor so 4 may play even with only 2 joystick ports.

Player response times may be selected, by the use of the option key, and may be of 3, 5 or 10 seconds in duration. A pause function must be used for unlimited response time. Each game consists of a series of rounds, with one round consisting of 10 questions. The preset option is for 5 rounds, but this may be altered to 8, 12, 16, or 20, making the games of from 50 to 200 questions in length.

The game may be played in one of two modes, competitive or social. In competitive mode all players or teams are competing at once to answer the displayed question. The first correct answer being the winner for the available points displayed. Social mode allows all the players to answer even after a correct answer has been given, of course more points are scored by the first correct answer.

Points are awarded by the use of a displayed time clock. The clock is set at 1000 points for multiple choice and 500 points for true/false. As the game proceeds the clock ticks off the points, the time of the correct answer determining its value. In PQ there is even a way for the various players to be handicapped. The handicapped player is given only half the time to answer a question. His controller is deactivated for the first half of the turn, not only allowing him half the time but reducing his point count by half.

Each controller is a simple hand held unit with four response buttons. The sets are attractive and well made. One caution is that the controller package must be in place prior to loading the game, if not some of the controllers may be inoperative. On the whole Suncomm has put together a nice little package. A little carelessness is evident in the questions. For example one question asks $7 \times 8 = 54$ true or false. It credits false as the correct answer but it tells you the answer is 58! Another gives "Ka" as the abbreviation for Kansas. Perhaps for a true trivia buff the lack of distinct categories might be disappointing, but as a general trivia game this one is good.

The program provides custom designed graphics screens, and there is some sound. The scores of top players are recorded and titles are given for attaining performance levels.

Suncomm has additional question disks for \$25. I want to see them issue a disk allowing you to set up your own series of questions. Without these additions, it is obvious at some point the available questions will have all been memorized.

— Nick Chrones

The ERACE Education disk #8 is now ready. These 6 programs cover a wide range of subjects. The first 5 programs are from John R. Kelley.

DOMINOS, DOMINOS2 and DOMINOS3 are separate, but chained arithmetic programs based on the spots of the double six, the double nine, and the double twelve dominos.

FLASH SPELL provides practice in recognizing and spelling over 300 common words.

FLASH READ helps to increase reading speed. A hundred short phrases are available.

CELL 12 is a science program from Paul Freeman teaching the basics of the cell. Very good graphics. Joystick and 32k are required.

This disk or any ERACE disk can be ordered from Ron Ness, ACE Librarian (374 Blackfoot, Eugene, OR 97404), for \$10. Double sided disks (with another library disk) are \$15.

Our next disk is half full. We need more education programs which are projects for your children, students, friends, and/or grandchildren. We can also use public domain programs. Send the program on disk or tape with documentation including your name and address.

We are still soliciting software for review. Even though we want to see as much commercial work as possible, we are just as interested in seeing your pet project in sharing ideas and knowledge.

If you are interested in joining ERACE, either call or drop a note into the mail. We will have meetings in Eugene with enough interested members.

— Nora Young
ERACE SIG

GETTING READY TO READ AND ADD

(\$40, Sunburst Communications. Recommended for ages 4-7)

This package includes six programs designed to give children practice in discriminating shapes (BEAM UP), upper and lower case letters (ALPHASURUS, ABC =, & LETTER GETTER), numerals (MOON MATH & NUMBER CHICK), and eye-hand coordination.

Correct responses cause novel and interesting things to happen on the screen with lively animation, colorful graphics, and sound. Incorrect responses are ignored.

These games can be changed by the user to control things like how fast, how many, and in what order information is presented. The user manual is very good. Each game has its own page along with a large picture, description, skills, change option, and objectives. There are suggestions for parents and teachers on how to present the program.

The disk is next to impossible to crash since one must press Control and the E key together to end the game.

We find the games to be a charming approach to teaching simple concepts. The graphics and sound, along with the change option make the games more interesting for the child. Katie, our first grader, enjoys the games and the rewards she gets for a good job. Joe, our 3rd grader, enjoys the graphics and effects.

— Nora Young

RGB FOR THE 800

(reprint: Bay Area Atari Users Group, March 1985)

SUBJECT: ATARI RGB OUTPUT

Atari 800 owners can now have an RGB output for \$20 or less by connecting to the A103 chip on the PC board under the metal plate on the bottom of the Atari 800.

For those with RGB adjustments on their monitor the installation is simple. The soldering of the five wires to the appropriate pins of the A103 chip and the soldering of the five leads to a chassis mounted 5-pin female RCA plug finishes the job (Fig. 1).

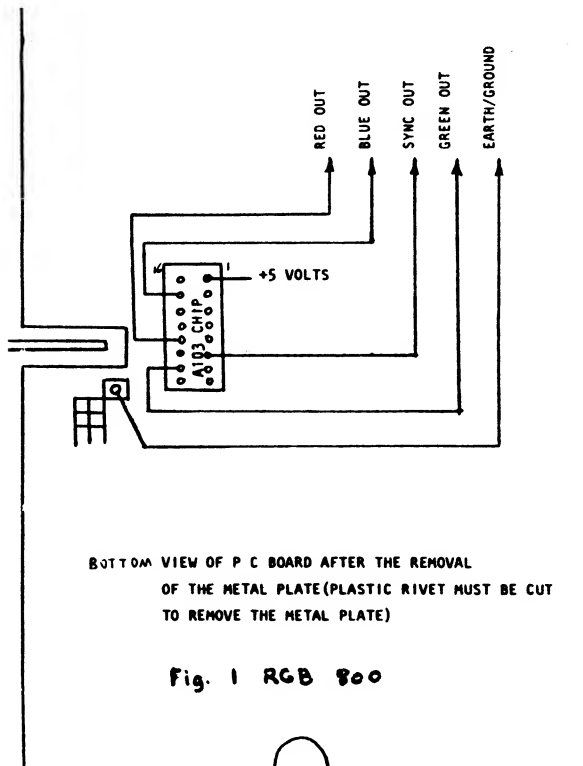
For all other owners of just the simple RGB video input with no separate RGB controls you just follow the schematic (Fig. 2) and make the same soldering connections as in the first example.

We find it is easier to use a 14 or 16 pin DIP socket to solder all four transistors on with the 68 ohm the 500 ohm (sic) potentiometers. This way you can do a clean installation by drilling three small holes in the side of the chassis for mounting the three potentiometers with a good two part epoxy so you can have easy adjustments of your RGB output with a small screwdriver. Then at the same time you can epoxy the 14 or 16 pin DIP socket to the inside of the 800 cabinet.

The total assembly and mounting time is under one hour.

PARTS LIST

- 4 2N2222 transistors
 - 4 68 ohm resistors
 - 3 500 ohm potentiometers
 - 1 14 or 16 pin DIP socket
 - 1 RCA 5 pin chassis mount female plug for appropriate RGB connection
- Plus some wire.



Atari Computer Enthusiasts

A.C.E. is an independent, non-profit and tax exempt computer club and user's group with no connection to the Atari Company, a division of Warner Communication Company. We are a group interested in educating our members in the use of the Atari Computer and in giving the latest News, Reviews and Rumors.

All our articles, reviews and programs come from you, our members.

Our membership is world-wide; membership fees include the A.C.E. Newsletter. Dues are \$14 a year for U.S., and \$24 a year Overseas Air-mail and include about 10 issues a year of the ACE Newsletter.

Subscription Dep't: 3662 Vine Maple Dr., Eugene, OR 97405.

**President— Robert Browning, 90 W. Myoak Dr., Eugene, OR 97404 503-689-1513

Vice Pres— Larry Gold, 1927 McLean Blvd., Eugene, OR 97405 503-686-1490

Secretary— Bruce Ebling, 1501 River Loop #1, Eugene, OR 97404 503-688-6872

Librarian— Ron and Aaron Ness, 374 Blackfoot, Eugene, OR 97404 503-689-7106.

Editors—
Mike Dunn, 3662 Vine Maple Dr., Eugene, OR 97405 503-344-6193
— Jim Bumpas, 4405 Dillard Rd., Eugene, OR 97405 503-484-9925
E.R.A.C.E. (Education SIG Editor)— Nora Young, 105 Hansen Lane, Eugene, OR 97404 503-688-1458

Send 50c stamps or coin (\$1 overseas) to the Ness' for the new, updated ACE Library List-new in May 84 !

Best of ACE books

Volume 1 contains bound issues of the ACE Newsletter from the first issue, Oct 81 to June of 1982

Volume 2 covers July 1982 to June 1983

Only \$12 each (\$2 extra for Airmail). Available only from:

George Suetsugu
45-602 Apuapu St
Kaneohe, HI 96744

SortFinder 1.2

A composite index of Atari related articles from 5 popular computer periodicals from Apr '81 to June '83, including ACE. Only \$6 for ACE member from:

Jim Carr, Valley Soft
2660 S.W. DeArmond
Corvallis, OR 97333

TYPESETTING FROM YOUR COMPUTER

ATARI OWNERS: If you have a modem, text editor, and communications program to send ASCII files, you should consider the improved readability and cost savings provided by **TYPESETTING** your program documentation, manuscript, newsletter, or other lengthy text instead of just reproducing it from line printer or daisy-wheel output. Computer typesetting by telephone offers you high quality, space-saving copy that creates the professional image you want! Hundreds of type styles to choose from with 8 styles and 12 sizes "on line." And it's easy to encode your copy with the few typesetting commands you need.

COMPLETE CONFIDENTIALITY GUARANTEED

— Bonded for your protection —

PUBLICATION DESIGN, EDITING, & PRODUCTION

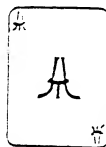
Editing & Design Services

30 East 13th Avenue Eugene, Oregon 97401
Phone 503/683-2657

Bulletin Board

(503) 343-4352

On line 24 hours a day, except for servicing and updating. Consists of a Tara equipped 48K Atari 400 with a TARA keyboard, 2 double-density double sided disk drives with an ATR 8000 interface, 2- 8" double density disk drives, an Epson MX80 printer, a Hayes SmartModem; running the ARMUDIC Bulletin Board software written by Frank L. Hubbard, 1206 N. Stafford St., Arlington, VA 22201. See the Nov '82 issue for complete details.



ATARI
COMPUTER
ENTHUSIASTS

3662 Vine Maple Dr. Eugene OR 97405

FIRST CLASS MAIL



Black Heritage USA 22